Concrete vs. Symbolic Simulation to Assess Cyber-Resilience of Control Systems

Giuseppina Mùrino, Armando Tacchella

KEYWORDS

Simulation of Control Systems, Artificial Intelligence, Cyber-Security and Critical Infrastructure Protection

ABSTRACT

State-of-the-art industrial control systems are complex implements featuring different spatial and temporal scales among components, multiple and distinct behavioral modalities, context-dependent and human-in-the-loop interaction patterns. Most control systems offer entry-points for malicious users to disrupt their functionality severely, which is unacceptable when they are part of the national critical infrastructure. Cyber-resilience, i.e., the ability of a system to sustain — possibly malicious — alterations while maintaining an acceptable functionality, is recognized as one of the keys to understand how much damage can be brought to a system and its surrounding environment in case of a successful cyber-attack. In this paper we compare methods to assess resilience considering both concrete simulation and symbolic simulation. Our ultimate goal is to provide maintainers and other stakeholders with a dynamic and quantitative measure of cyber-resilience. Here we present some results on a case study related to waste-water treatment, in order to provide initial evidence that concrete and symbolic simulation can be used in a complementary way to analyze the security of industrial control systems.

INTRODUCTION

When considering industrial control systems, one may observe that current state-of-the-art systems are complex implements intertwining physical processes, hardware, software, and communication networks — the term *cyber-physical system* (CPS) is often used in this context. With respect to "classical" embedded systems, a CPS adds elements of complexity including different spatial and temporal scales among components, multiple and distinct behavioral modalities, context-dependent and human-in-the-loop interaction patterns [Lee08]. Examples of CPSs include heterogeneous systems of systems such as water treatment plants, electric grids (power plants and associated distribution networks), industrial plants, transportation vehicles, and smart homes.

Wireless communication among components and external network access for supervisory control and data acquisition (SCADA) make any control system an ideal target for cyber-attacks. It has been demonstrated that many such systems offer potential entry-points for malicious users to gain control of the controlled system and/or disrupt its functionality severely. This is true also of most CPSs which are part of national critical infrastructure (CI) and, as such, intentional or accidental incidents that alter the regulation of their parameters, feedback lines and/or set points can have dramatic effects on the safety of citizens [WFD10].

Among other security-related issues, resilience is recognized as one of the keys to understand how much damage can be brought to a system and its surrounding environment in case of a successful cyber-attack [DRKS08]. The concept of resilience — literally defined as "*the capacity to recover quickly from difficulties; toughness*" or "*the quality of being able to return quickly to a previous good condition after problems*" — emerges as an additional target, complementary to protection from external threats, but not subordinate to it. This line of thought is pervasive in the Presidential Policy Directive 21 [Oba13] about CI security, which defines resilience as "*[...] the ability to [...] withstand and recover rapidly from disruptions. Resilience includes the ability to withstand and recover from deliberate attacks, accidents, or naturally occurring threats or incidents*". More recently, the term *cyber-resilience* has been coined to identify specifically "*the ability to continuously deliver the intended outcome despite adverse cyber events*" [BHSZ15], and this is the interpretation whereto we adhere in this paper.

We are interested in a quantitative and succinct measure of resilience, i.e., one that describes as precisely as possible the amount of damage that a system can tolerate before becoming unstable or exhibit undesirable and potentially dangerous behaviors. Mostly outside of cyber-attack scenarios, resilience is a well-studied topic from a control-theoretic standpoint [RKC09], [Far15], [WMW+16], but "classical" approaches are limited to relatively small system components with tractable dynamics. Here we consider simulation techniques to bridge the gap from single components to the system as a whole, and deal with nonlinear and hybrid dynamics. We rely on industry-standard Matlab/Simulink® software to provide modeling and simulation capabilities, and we consider Monte-Carlo style simulations — henceforth referred to as *concrete simulation* — to assess the envelope of system responses with and without attacks and quantify resilience as a ratio of the envelopes. In spite of its expressive power, concrete simulation is often insufficient to foresee all the potential effects of alterations brought by cyber-attacks because the combinatorial explosion of potential states. We propose to complement concrete simulation with algorithmic techniques which, without executing the system, analyze it to find violations of stated requirements — henceforth refereed to as *symbolic simulation*. The approach we consider is Model Check-

Giò Mùrino and Armando Tacchella are with "Dipartimento di Informatica, Bioingegneria, Robotica e Ingegneria dei Sistemi" (DIBRIS), University of Genoa, Viale Causa 13, 16145 Genoa, Italy. E-mail: giuseppina.murino@edu.unige.it, armando.tacchella@unige.it. The authors wish to thank Leonardo S.p.A. for supporting the research. The corresponding author is Armando Tacchella.

Proceedings 32nd European Conference on Modelling and Simulation ©ECMS Lars Nolle, Alexandra Burger, Christoph Tholen, Jens Werner, Jens Wellhausen (Editors)
ISBN: 978-0-9932440-6-3/ ISBN: 978-0-9932440-7-0 (CD)

ing [BKL08], i.e., given the model of a system and a property to check, prove whether the system upholds the property or not. While model checking is well known in hardware and software verification, its application to CPSs is not main-stream yet, so it is fruitful to compare it vs. standard (concrete) simulation techniques.

The main contribution of this paper is twofold. From the methodological point of view, we propose to merge approaches of concrete and symbolic simulation to evaluate the resilience of a system. Here, the goal is to provide maintainers and other stakeholders with a dynamic and quantitative measure of the resilience of a system. From the engineering point of view, we compare both methodologies on a case study related to waste-water treatment. The objective is to provide preliminary evidence that national critical infrastructure security can be enhanced considering our approach both in safety-by-design or safety-by-retrofit frameworks. The rest of the paper is structured as follows. In Section "HYBRID AUTOMATA" we introduce basic terminology and definitions related to the mathematical model we consider for CPSs. In Section "CASE STUDY" we introduce our case study related to a waste-water treatment facility and we describe the model we consider. In Section "CONCRETE VS. SYMBOLIC SIMULATION" we outline the two approaches. In Section "EXPERIMENTAL EVALUATION", we present some results related to the facility herewith described and we conclude the paper in with some final remarks in Section "CONCLUSIONS".

HYBRID AUTOMATA

In order to model CPSs we resort to the formalism of *Hybrid Automata* [ACHH93]. For our purposes, a hybrid automaton can be defined as a tuple $A = (X, V, flow, inv, init, E, jump)$ consisting of the following components:

Variables are a finite ordered set $X = \{x_1, x_2, \dots x_n\}$ of real-valued variables, representing the continuous component of the system's state.

Control modes are a finite set V, representing the discrete component of the system's state.

Flow conditions are expressed with a labeling function $flow$ that assigns a condition to each control mode $v \in V$. The flow condition $flow(v)$ is a predicate over the variables in $X \cup \dot{X}$, where $\dot{X} = \{\dot{x}_1, \dot{x}_2, \dots, \dot{x}_n\}$. The dotted variable $\dot{x}_i$ for $1 \leq i \leq n$ refers to the first derivative of x_i with respect to time, i.e., $\dot{x}_i = \frac{dx}{dt}$.

Invariant conditions determine the constraints of each control mode with the labeling function inv, and *initial conditions* are denoted with the function $init$.

Control switches are a finite multiset $E \in V \times V$. Each control switch (v, v') is a directed edge between a source mode $v \in V$ and a target mode $v' \in V$.

Jump conditions are expressed with a labeling function $jump$ that assigns a jump condition to each control switch $e \in E$. The jump condition $jump(e)$ is a predicate over the variables in $X \cup X'$, where $X' = \{x'_1, x'_2, \dots x'_n\}$. The unprimed symbol x_i, for $1 \leq i \leq n$, refers to the value of the variable x_i before the control switch, and the primed symbol x'_i refers to the value of x_i after the control switch. Thus, a jump condition relates the values of the variables before a control switch to the possible values after the control switch.

Intuitively, the evolution of a hybrid automata can be associated to a sequence of transition between "locations" characterized by specific control modes and values of the variables. In order to frame this concept precisely we define a *state* as a pair $(v, \mathbf{a})$ consisting of a control mode $v \in V$ and a vector $\mathbf{a} = (\mathbf{a_1}, \dots, \mathbf{a_n})$ that represents a value $a_i \in \mathbb{R}$ for each variable $x_i \in X$. The state $(v, \mathbf{a})$ is *admissible* if the predicate $inv(v)$ is true when each variable x_i is replaced by the value a_i. The state $(v, \mathbf{a})$ is *initial* if the predicate $init(v)$ is true when each x_i is replaced by a_i. Consider a pair of admissible states $q = (v, \mathbf{a})$ and $q' = (v', \mathbf{a}')$. The pair (q, q') is a *jump* of A if there is a control switch $e \in E$ with source mode v and target mode v' such that the predicate $jump(e)$ is true when each variable x_i is replaced by the value a_i, and each primed variable x'_i is replaced by the value a'_i. The pair (q, q') is a *flow* of A if $v = v'$ and there is a non-negative real $\delta \in \mathbb{R}_{\geq 0}$ – the duration of the flow – and a differentiable function $\rho : [0, \delta] \to \mathbb{R}^{\kappa}$ – the curve of the flow – such that (i) $\rho(0) = \mathbf{a}$ and $\rho(\delta) = \mathbf{a}'$; (ii) for all time instants $t \in (0, \delta)$ the state $(v, \rho(t))$ is admissible; and (iii) for all time instants $t \in (0, \delta)$, the predicate $flow(v)$ is true when each variable x_i is replaced by the i-th coordinate of the vector $\rho(t)$, and each $\dot{x}_i$ is replaced by the i-th coordinate of $\dot{\rho}(t)$ – where $\dot{\rho} = \frac{d\rho}{dt}$. In words, jumps define the behavior of the automaton when switching from one control mode to another, whereas flows describe the behavior of the automaton inside the control mode. With the concepts above, we can now define executions of the automaton as *trajectories*, i.e., finite sequences $q_0, q_1, \dots, q_k$ of admissible states q_j such that (i) the first state q_0 of the sequence is an initial state of A, and (ii) each pair $(q_j, q_j + 1)$ of consecutive states in the sequence is either a jump of A or a flow of A. A state of A is *reachable* if it is the last state of some trajectory. For the purpose of this paper, the analysis of a hybrid system, amounts to computing the set of reachable states.

Given the expressiveness of the above formalism, it is no surprise that evaluating the reachability of given (sets of) states is, in its most general form, an undecidable problem in hybrid automata [ACHH93]. Even if the control system can be modeled in terms of a *linear hybrid automaton*, i.e., a hybrid automaton where the dynamics of the continuous variables are defined by linear differential inequalities, there is still no guarantee that the exploration of the set of reachable states terminates. The method is still of practical interest, however, because terminations can be enforced by considering the behavior of the system over a bounded interval of time. This technique, known as Bounded Model Checking – see, e.g., [FHT$^+$07] – involves the exploration of increasingly long trajectories until either an unsafe state is reached, or resources (CPU time, memory) are exhausted. Technically, we no longer speak of verification, which is untenable in an infinite state space, but of falsification. Whenever an unsafe state is found, then we have a trajectory witnessing the bug. On the other hand, the unfruitful exploration of increasingly long trajectories is considered an empirical guarantee of safety.

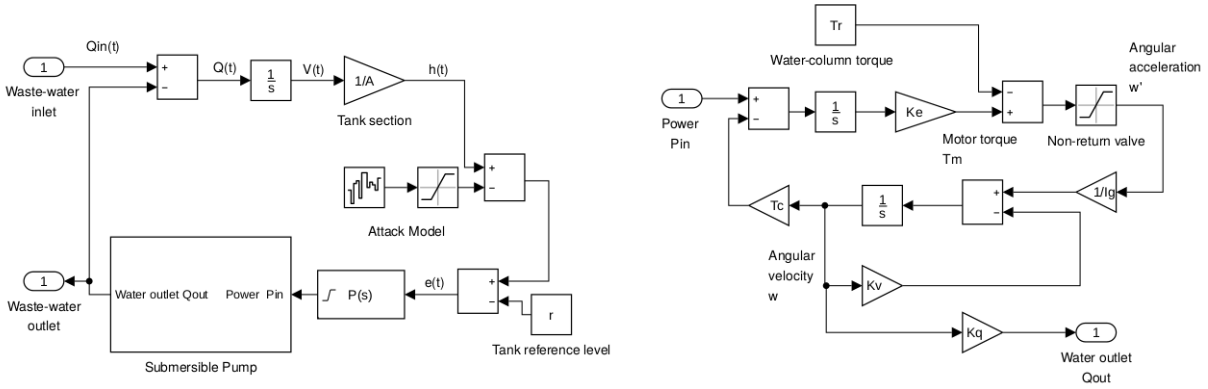

Fig. 1: Model of the nitrification-oxidification tank control system (left) and detail of the submersible pump (right).

CASE STUDY

Our analysis focuses on a waste-water treatment facility whose name and location cannot be disclosed, but whose features and data were made accessible to us to perform this study. In order to achieve a realistic, yet manageable case study, we decided to model only the main waste-water cycle. Components that we did not model include, for instance, sludge treatment and air-quality control. The water cycle is comprised of two main compartments: In the first compartment (pre-treatment), waste-water pumped from the city sewage network is filtered to remove coarse debris, then it is stationed in a tank to remove sand and oil/fat, and finally it is conveyed to a buffer towards further treatment. In the second compartment (biological), waste-water undergoes a denitrification process, then a nitrification-oxidation process, and finally it is conveyed through micro-membranes (MBR) before being released. The subject of our study is the tank wherein nitrification-oxidation (NO) process is carried out. In Figure 1 (left) we show the detailed Matlab/Simulink® model of the NO tank. As we can see from the schematics, we have assumed a simplified linear model for the tank dynamics, whereby the total volume $V(t)$ of fluids contained in the tank is obtained by integrating the net inlet $Q(t)$ which, in turn, is obtained by subtracting the output flow $Q_{out}(t)$ from the tank inlet $Q_{in}(t)$. While the latter is an input to the NO tank, the tank outlet is controlled by electrical pumps driven by a proportional regulator tracking a given reference level r — detail of the motor/pump model is given in Figure 1 (right). The goal of the regulator is to avoid the tank becoming too full, so as to avoid triggering emergency bypasses, or too empty, so as to avoid impairing the chemical process undergoing in the NO tank. Both events are undesirable because bypasses dump to the sea untreated sewage liquor, whereas incomplete chemical processing of waste-water may cause failures in subsequent steps.

We consider the potential modifications that an attacker can bring by gaining access to the system. Since, from a theoretical point of view, the essential mechanism in every CPS is the feedback control loop, an attacker gaining access to the control system can alter the system in three ways only: (a) by changing the set point, (b) by altering the feedback signal, and (c) by changing the regulator parameters. Consider, as an example, the control loop that keeps the level of the tank close to the desired level shown in Figure 1 (left). Here, attack (a) corresponds to changing the desired tank level, attack (b) corresponds to altering the actual tank level, and attack (c) corresponds to changing the proportional gain of the regulator. In practice, an attacker may decide to perform all such actions and in more than one part of the system, as well as other disruptive actions like blocking the functionality of components, e.g., shutting down control devices or flooding them with requests in order to hamper their functionality. As shown in Figure 1 (left), in our simulation we have considered an attack model wherein the feedback signal from the tank is altered — attack (b) — which, in our hypothesis of additive modification, makes it equivalent to alteration of the set point — attack (a). We did not consider attack (c) as well as the possibility of multiple or blocking attacks. The attacker is simulated considering the injection of a faulty signal using a band-limited white noise (BLWN) block, saturated to give output in the range [-1;1]. The idea behind this model is that alterations of the set point or the feedback signal greater than one meter could trigger anomaly alarms — e.g., those guarding against sensor malfunction — but within these bound the attacker can freely modify the feedback value. The impact of the attacker can be regulated by considering the noise power and sample time parameters of the BLWN block. In particular, noise power correlates with the "strength" of the attack, whereas sample time correlates with the "frequency" of the attack. We focus our study of cyber-resilience on the NO tank control system, in the hypothesis that an attacker may gain virtual access to the facility network and act according to the hypotheses above. It is worth noticing that this kind of attack is the same staged by the famous Stuxnet virus [FR11] and therefore, independently from the likelihood of an attacker penetrating the system, it is worth evaluating the impact of such a threat in terms of cyber-resilience.

CONCRETE VS. SYMBOLIC SIMULATION

To be defined as such, an attack must alter the normal behavior of the system in order to cause damage or undesirable effects, which we define as follows:

- Increased frequency of daily power duty cycles to the pump; we consider as a measure of this effect the daily av-

erage of power peaks, defined as the events such that the power delivered to the pump reaches above 80% of the nominal power.

- Increased span of variation in power delivery to the pump; we consider as a measure of this effect the interquartile range[1] of the regulator output signal which in our model corresponds to power delivery to the pump.
- Overflow in the NO tank; technically, such overflows may never happen in practice because of emergency bypasses, but in our model we use the overflow condition to identify when bypasses should be opened.
- Underflow in the NO tank; technically, an underflow in the NO tank may happen in practice due to decreased inlet from the network and excessive outlet of liquor.

All the conditions above may cause damage to the plant or to the surrounding environment to some extent. In particular, the first two conditions may severely reduce the useful life of pump motors and increase the chance of pump failures; the third condition may cause the opening of the bypasses and the dumping of partially-treated liquor at sea; finally, the fourth condition may alter the chemical-biological process, hamper the liquor purification process and cause damage in the NO tank.

As mentioned in the introduction, we consider two different approaches to assess resilience under attack. In particular, starting from the model outlined in the previous section, we wish to compare:

- Empirical analysis of resilience through concrete simulation: since the model in Figure 1 is "executable", this technique is readily available. Its advantages are simplicity, expressiveness and iterative improvability, i.e., running more simulation will yield more precise results. Its main disadvantage is that, due to the combinatorial explosion of inputs, it is impossible to test all potential system configurations to quantify resilience in a precise way.
- Comparing reachable sets through symbolic simulation, i.e., bounded model checking of hybrid automata: given a hybrid automaton modeling the relevant behaviors of the system, it is possible to perform reachability analysis, i.e., (over)estimate the trajectories of the system without actually executing it as required by simulation. Advantages and disadvantages of symbolic simulation are complementary to concrete simulation: since symbolic simulation considers an over-approximation of the reachable sets, no trajectory is possibly left out. However, the computational cost of symbolic simulation is usually much higher than concrete simulation.

As a yardstick for the application of both methodologies, here we consider concrete simulation. In particular, simulation of the plant is performed with and without assuming external attack attempts, and we consider historical data made available from the managing utility to simulate sewage inlet. In order to get meaningful sampling of the potential input space, we consider a daily inlet profile under conditions of maximum utilization - and obtain random variates of such basic profile by adding (band-limited) Gaussian white noise which provides random but realistic deviations from the historical profile. In the following, we call *baseline scenario* the simulation obtained by running the plant without attack, and we define an *attack scenario* where alternations are small, but persistent with the goal of going unnoticed, but still damage the plant in some way. In order to quantify resilience decrease under attack conditions we compare the envelope of the regulator output to the one in the baseline scenario with the following formula

$$R = \frac{\max(P(t))\min(P(t))}{\max(P_a(t))\min(P_a(t))} \tag{1}$$

where $P(t)$ is the regulator power signal in the baseline simulation, and $P_a(t)$ is the same signal in the attack scenario.

To perform resilience evaluation using model checking of hybrid automata we must manually compile, for each control mode, a state equation of the form $\dot{x} = Ax + Bu$, where the vector $x \in \mathbb{R}^n$ represents the state of the system, $u \in \mathbb{R}^m$ is the system input or disturbance, and A, B are matrices of appropriate size. The state equations of specific modes plus (a) a set of potential initial states and (b) boundary conditions for inputs/disturbances has to be supplied to the model checker. While (a) poses no problems and (b) can be shaped in order to take into account the hypothesis of attacks of various intensity brought to the system, the main issue is that the system described in Figure 1 requires at least five control modes to take into account the presence of saturations: one in the pump feedback loop, and another in the motor model. The former saturation is due to the fact that the regulator does not emit "negative" power and does not drive the pump at more than 15 kW — the nominal rating of the pump. The latter saturation is due to the fact that a non-return valve prevents the pump to spin backwards when the motor torque is less than the resistant torque expressed by the water column above the pump. Accounting for these effects requires modeling the system using a hybrid automaton.

When no saturation effect is present, the differential equations modeling the system in Figure 1 are the following:

$$\begin{cases} \dot{h}(t) &= -\frac{K_q}{S}w(t) + \frac{1}{S}Q_{in}(t) \\ \dot{T}_m(t) &= K_e K_{PID} h(t) - K_e T_c w(t) + \\ & \quad -K_e K_{PID}(r(t) + N(t)) \\ \dot{w}(t) &= \frac{1}{I_g}T_m(t) - \frac{1}{K_v}w(t) - \frac{1}{I_g}T_r(t) \end{cases} \tag{2}$$

where:

- $h(t)$ is the height of the liquor in the NO tank: the inlet flow $Q_{in}(t)$ minus the outlet flow, which is a constant K_q multiplied by the rotational speed of the pump $w(t)$, both divided by the area of the tank S, determine the variation of the tank height.
- $T_m(t)$ is the torque of the motor powering the pump: to obtain it, we consider $e(t) = h(t) - r(t) - N(t)$ — called *error signal* in the following; the variation of T_m is influenced by two terms: the product of $e(t)$, the proportional gain of the controller K_{PID}, and the power transfer K_e; the second term is $K_e T_c w(t)$, i.e., a quantity proportional to the speed of the motor which represents the tendency of asynchronous motors to decrease torque when approaching synchronous speed.

[1] The interquartile range of a distribution is the difference between its 75^{th} percentile and 25^{th} percentile. As such, it provides a measure of the dispersion of the distributions values, one which is more resistant than standard deviation to the presence of outliers.

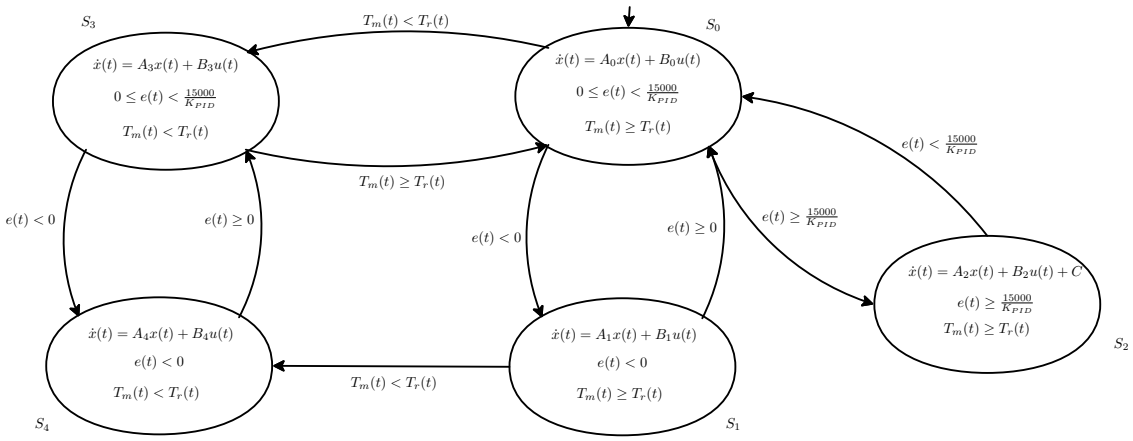

Fig. 2: Hybrid automaton representing the NO tank control system with saturations in power delivery and motor pump. Each oval represents a control mode, i.e., a discrete state S_i with $i \in \{0, 1, 2, 3, 4\}$. Inside the ovals, the flow is described by an ordinary differential equation in matrix form (variables as defined in the text). The invariants are the predicate appearing below the differential equation. Each edge is labeled with the corresponding jump condition, i.e., when the condition becomes true, the control mode is changed. The initial state is denoted with an incoming arrow.

- $w(t)$ is the rotational speed of the pump: the corresponding angular acceleration $\dot{w}$ is determined by the difference between the motor torque T_m on one side, and the resistant torque T_r — due to the water column above the pump — plus the torque due to viscous friction $\frac{w(t)}{K_v}$ on the other side; here, I_g is the rotational inertia of the pump rotor and K_v is the viscous friction coefficient.

If $N(t) = 0$ for all $t \in (0, +\infty)$, then the equations represent the baseline scenario, otherwise $N(t)$ represents the attacker interfering with the set point $r(t)$ in the attack scenario. Let us now consider the *state vector* as the (column) vector $x(t) = [h(t), T_m(t), w(t)]$ and the *input vector* as the (column) vector $u(t) = [Q_{in}(t), r(t) + N(t), T_r(t)]$. Equation (2) can be rewritten as $\dot{x} = A_0x + B_0u$ where A_0 and B_0 are 3×3 matrices:

$$A_0 = \begin{bmatrix} 0 & 0 & -\frac{K_q}{S} \\ K_eK_{PID} & 0 & -K_eT_c \\ 0 & \frac{1}{I_g} & -K_v \end{bmatrix} \tag{3}$$

$$B_0 = \begin{bmatrix} \frac{1}{S} & 0 & 0 \\ 0 & -K_eK_{PID} & 0 \\ 0 & 0 & -\frac{1}{I_g} \end{bmatrix} \tag{4}$$

These equations correspond to the control mode S_0 in Figure 2 where the invariants $0 \le e(t) < \frac{15000}{K_{PID}}$ and $T_m(t) \ge T_r(t)$ are both satisfied, i.e., no saturation effect is present. The other control modes are obtained considering what happens to equation (2) when the invariants above are not satisfied.

In particular, when the error signal $e(t)$ is negative, the power delivered to the motor pump is 0. This dynamic can be described by a system of the form $\dot{x} = A_1x + B_1u$ where the matrices A_1 and B_1 are defined as follows:

$$A_1 = \begin{bmatrix} 0 & 0 & -\frac{K_q}{S} \\ \mathbf{0} & 0 & -K_eT_c \\ 0 & \frac{1}{I_g} & -K_v \end{bmatrix} \tag{5}$$

$$B_1 = \begin{bmatrix} \frac{1}{S} & 0 & 0 \\ 0 & \mathbf{0} & 0 \\ 0 & 0 & -\frac{1}{I_g} \end{bmatrix} \tag{6}$$

As we can see, in both A_1 and B_1 the coefficient due to power transfer from the controller to the pump (K_eK_{PID}) is now 0 (in bold). In Figure 2 this corresponds to control mode S_1. On the other hand, whenever $e(t) \ge \frac{15000}{K_{PID}}$, then the power delivery to the motor of the pump is capped at $K_C = 15$ kW. The corresponding dynamic can now be described by a system of the form $\dot{x} = A_2x + B_2u + C$ where C is the (column) vector $C = [0, K_eK_C, 0]$, $A_2 = A_1$ and $B_2 = B_1$; this configuration corresponds to control model S_2 in Figure 2. Notice that control modes S_0, S_1 and S_2 all satisfy the invariant $T_m(t) \ge T_r(t)$. In principle, in each such state this invariant might not hold, which would result in an automaton with six states. However, in state S_3 since the motor is operating at nominal power, it is not possible to reach a configuration where $T_m(t) < T_r(t)$ unless there is a problem in the pump outlet, e.g., a clog, which is not the subject of our attention.

Control modes S_3 and S_4 in Figure 2 correspond to configurations in which $T_m(t) < T_r(t)$. In case of S_3 this happens when the power signal delivered to the motor is not strong enough to overcome the resistant torque. The matrices corresponding to this mode are:

$$A_3 = \begin{bmatrix} 0 & 0 & -\frac{K_q}{S} \\ K_eK_{PID} & 0 & -K_eT_c \\ 0 & \mathbf{0} & -K_v \end{bmatrix} \tag{7}$$

$$B_3 = \begin{bmatrix} \frac{1}{S} & 0 & 0 \\ 0 & -K_eK_{PID} & 0 \\ 0 & 0 & \mathbf{0} \end{bmatrix} \tag{8}$$

As before, we highlight in boldface the terms which differ with respect to matrices A_0 and B_0 in equations (3) and (4). In this control mode, the pump (if moving) is bound to stop because of the viscous friction. Additionally, is $e(t) < 0$ then the configuration modeled by control mode

NOISE POWER	SAMPLE TIME	NO tank level [m]	MBR tank level [m]	PID input	Power IQR [KW]	Power Peaks (daily average)	MBR input flow [m^3/s]
	60	3.5 - 4.3	0.3 - 1.8	OK	5.118	3431.9	OK
100	600	3.7 - 5.25	0 - 3.4	1.6	3.658	5849.2	0.32
	6000	4.3 - 5	0 - 2.7	OK	0.984	89.24	0.305
	60	3.5 - 4	0.4 - 1.6	OK	5.369	109.23	OK
1000	600	3.5 - 5.3	0 - 3.8	2	3.381	6983.9	0.32
	6000	3.5 - 5.62	0 - 3.8	1.7	1.193	1941.2	0.32
	60	3.7 -4.10	0.3 - 1.8	OK	5.369	109.23	OK
10000	600	3.5 - 5.3	0 - 4.10	1.8	3.610	4690.3	0.32
	6000	3.5 - 5.62	0 - 4.51	2.12	2.841	4331.2	0.32

Fig. 3: Results of concrete simulation in the attack scenario.

S_4 is reached. Here, no power is delivered to the motor and no acceleration is given to the pump. Matrices A_4 and B_4 can be obtained by A_1 and B_1 by zeroing the same entries highlighted in (7) and (8). The only dynamics left in S_4 are those related to the tank filling by effect of the inlet Q_{in}. Notice that, in this configuration it is impossible for $T_m(t)$ to become greater than $T_r(t)$. Therefore, the only possibility is for $e(t)$ to become larger than 0 and then large enough to make $T_m(t) > T_r(t)$ again, i.e., move from control model S_4, back to S_3 and then S_0.

EXPERIMENTAL EVALUATION

In Figure 3 we present a table with the results obtained by simulating the undercover attack scenario in 9 different configurations of the BLWN block simulating the attacker. Simulations run across 100 days starting from regime conditions — corresponding to control mode S_0 in Figure 2 — and statistics are reported on a daily basis. The table is organized as follows. Columns "NOISE POWER" and "SAMPLE TIME" report the corresponding parameters of the BLWN block; columns "NO tank level" and "MBR tank level" report the minimum and maximum levels reached during simulation by the NO and MBR tanks, respectively — the MBR tanks are next to the NO tank, and they are critical for the whole process;"PID input" reports about the error signal $e(t)$ entering the regulator: OK means that the signal is contained within the range $[-4.5; +1]$, a value means that the range was exceeded up to that value; "Power IQR" and "Power Peaks" report about regulator output: "IQR" is the interquartile range of the power signal, and "Peaks" are the number of times in which 80% of the nominal pump power is exceeded on a daily basis; finally, "MBR input flow" refers to the NO tank outlet which is also the MBR tank inlet: OK means that the inlet is less than 0.3 cubic meters per second, a value means that the threshold was exceeded up to that value. From the table we draw the conclusion that all the simulations presented in the scenario qualify as attacks because they meet one or more of the conditions defined in the methodological section. The simulations with exceeding thresholds (red values) are clearly definable as such (and become overt attacks), whereas the simulations where the attack frequency is low (60 seconds of sample time) do not have a detectable impact, but more subtle effects instead. In particular, a baseline simulation without hacker features a power IQR of 0.874 KW, whereas in the simulations herewith considered the power IQR never falls below 5 KW when the sample time is 60 seconds; also the baseline scenario does not feature power peaks — power is consistently regulated to follow the tank inlet — whereas in the attack scenarios the average number of peaks can reach the order of thousands. Considering the simulations above, the value of the ratio in equation (1) is 0.13 in all the cases displayed in Figure 3. This means that, under attack conditions, resilience of the system is reduced to almost one tenth of the system working in normal conditions according to the simulation leg of our methodology. Similar results are obtained with other "sensible" signals, including motor torque and angular velocity.

Considering the techniques described in [ALGK11] and embodied in the tool CORA (COntinuous Reachability Analyzer)[2] we consider the model presented in Figure 2 and try to reach conclusions which are independent from the specific simulation. Modeling hybrid dynamics in CORA requires an effort which is beyond the scope of the paper. In order to get an idea of the extra capabilities granted by CORA — and model checking techniques in general — in Figure 4 we show the plot of reachable sets in the plane where the x-axis is the NO tank height and the y-axis is the motor torque considering control mode S_0 of the automaton depicted in Figure 2. As we can see, in the case of no attack brought to the system (plot on the left), the controller is able to maintain a satisfactory height of the tank, even if initial conditions where slightly off-equilibrium, and this is the case for all potential system trajectories from the set of initial states considered. On the other hand, if an attack is brought to the initial state (plot on the right), CORA shows that the system stabilizes again, but this is an artifact of the model since the motor torque, by construction, cannot exceed 30 Nm at maximum rated power. Albeit in a qualitative fashion, using CORA allows us to evaluate the impact of such attack and potentially quantify resilience once the correct dynamics can be taken into account. More precise

[2] http://www.i6.in.tum.de/Main/SoftwareCORA

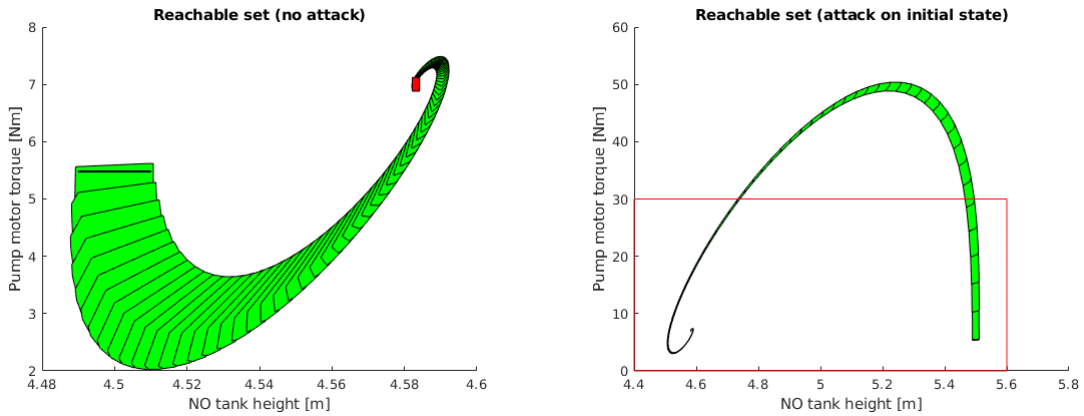

Fig. 4: Analysis of the system with CORA: baseline scenario (left) and attack scenario (right).

quantification can be obtained using a hybrid model, which will be the subject of further study.

CONCLUSIONS

We have shown that, considering a simplified but realistic case study, simulation is able to quantify the decrease of resilience in a system under attack, whereas model checking techniques have the potential to complement it providing further insights. As a future work, we plan to consolidate our methodology by further integrating its elements of the approach and by formalizing the theoretical connections between them. On the engineering side, we wish to extend our analysis to cover the whole waste-water treatment in the facility considered by providing more accurate modeling of various security-critical components. The result of this effort will enable better protection of the infrastructure, and will provide also the first seed of a tool to simulate attacks and responses in order to train facility personnel or test protection devices. We also plan to further validate our methodology by extending it to evaluate resilience of other critical-infrastructure facilities, with a focus on energy production plants and distribution networks.

REFERENCES

[ACHH93] R. Alur, C. Courcoubetis, T.A. Henzinger, and P.H. Ho. Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems. *Lecture notes in computer science*, pages 209–229, 1993.

[ALGK11] Matthias Althoff, Colas Le Guernic, and Bruce H Krogh. Reachable set computation for uncertain time-varying linear systems. In *Proceedings of the 14th international conference on Hybrid systems: computation and control*, pages 93–102. ACM, 2011.

[BHSZ15] Fredrik Björck, Martin Henkel, Janis Stirna, and Jelena Zdravkovic. Cyber resilience-fundamentals for a definition. In *WorldCIST (1)*, pages 311–316, 2015.

[BKL08] Christel Baier, Joost-Pieter Katoen, and Kim Guldstrand Larsen. *Principles of model checking*. MIT press, 2008.

[DRKS08] Salvatore DAntonio, Luigi Romano, Abdelmajid Khelil, and Neeraj Suri. Increasing security and protection through infrastructure resilience: the inspire project. In *International Workshop on Critical Information Infrastructures Security*, pages 109–118. Springer, 2008.

[Far15] Amro M Farid. Static resilience of large flexible engineering systems: Axiomatic design model and measures. *IEEE Systems Journal*, 2015.

[FHT+07] M. Franzle, C. Herde, T. Teige, S. Ratschan, and T. Schubert. Efficient solving of large non-linear arithmetic constraint systems with complex boolean structure. *Journal on Satisfiability, Boolean Modeling and Computation*, 1:209–236, 2007.

[FR11] James P Farwell and Rafal Rohozinski. Stuxnet and the future of cyber war. *Survival*, 53(1):23–40, 2011.

[Lee08] Edward A. Lee. Cyber physical systems: Design challenges. In *11th IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC 2008), 5-7 May 2008, Orlando, Florida, USA*, pages 363–369, 2008.

[Oba13] Barack Obama. Presidential policy directive 21: Critical infrastructure security and resilience. *Washington, DC*, 2013.

[RKC09] Dorothy A Reed, Kailash C Kapur, and Richard D Christie. Methodology for assessing the resilience of networked infrastructure. *IEEE Systems Journal*, 3(2):174–180, 2009.

[WFD10] Chunlei Wang, Lan Fang, and Yiqi Dai. A simulation environment for scada security analysis and assessment. In *Measuring Technology and Mechatronics Automation (ICMTMA), 2010 International Conference on*, volume 1, pages 342–347. IEEE, 2010.

[WMW+16] Junwei Wang, Raja R Muddada, Hongfeng Wang, Jinliang Ding, Yingzi Lin, Changli Liu, and Wenjun Zhang. Toward a resilient holistic supply chain network system: Concept, review and future direction. *IEEE Systems Journal*, 10(2):410–421, 2016.

GIUSEPPINA MURINO graduated from the University of Genoa, Italy with a "Laurea Magistrale" (MSc equivalent) in Management Engineering in December 2013. She is now a Research Engineer at the Department of Informatics, Bioengineering, Robotics and System Engineering, also pursuing PhD studies in the field of modeling and simulation. She is currently the technical leader of the project aimed at studying the resilience of cyber-physical systems controlling elements of national critical infrastructure.

ARMANDO TACCHELLA graduated from the University of Genoa, Italy with a "Laurea" (MSc equivalent) in Computer Engineering and a PhD in Computer Science and Engineering. Dr. Tacchella was a research associate at Rice University in Houston (US) and then a research fellow at University of Genoa, where he became associate professor of information processing systems in 2005. His research interests are in the field of artificial intelligence and formal methods applied to system engineering.