Design and Evaluation of Jamming Resilient Cyber-Physical Systems

Ivana Tomić, Michael J. Breza, Greg Jackson, Laksh Bhatia and Julie A. McCann
Department of Computing, Imperial College London, UK
Email: {i.tomic, michael.breza04, greg.jackson, laksh.bhatia16, j.mccann}@imperial.ac.uk

***Abstract*—There is a growing movement to retrofit ageing, large scale infrastructures, such as water networks, with wireless sensors and actuators. Next generation Cyber-Physical Systems (CPSs) are a tight integration of sensing, control, communication, computation and physical processes. The failure of any one of these components can cause a failure of the entire CPS. This represents a system design challenge to address these interdependencies. Wireless communication is unreliable and prone to cyber-attacks. An attack upon the wireless communication of CPS would prevent the communication of up-to-date information from the physical process to the controller. A controller without up-to-date information is unable to meet system's stability and performance guarantees. We focus on design approach to make CPSs secure and we evaluate their resilience to jamming attacks aimed at disrupting the system's wireless communication. We consider classic time-triggered control scheme and various resource-aware event-triggered control schemes. We evaluate these on a water network test-bed against three jamming strategies: constant, random, and protocol aware. Our test-bed results show that all schemes are very susceptible to constant and random jamming. We find that time-triggered control schemes are just as susceptible to protocol aware jamming, where some event-triggered control schemes are completely resilient to protocol aware jamming. Finally, we further enhance the resilience of an event-triggered control scheme through the addition of a dynamical estimator that estimates lost or corrupted data.**

***Index Terms*—Cyber-Physical Systems, Event-Triggered Control, Wireless Sensor/Actuator Networks, Security, Jamming.**

I. Introduction

There is a recent trend to instrument and control large infrastructure installations like water distribution networks (WDNs), precision agriculture or unmanned off-shore oil rig systems with wireless sensor and actuator networks [1]. In this paper we present a design approach to make these Cyber-Physical Systems (CPSs) secure to jamming attacks. We assess the robustness of different controllers, as well as other qualities such as energy efficiency on a physical test-bed. The result of our design approach is the most jamming resilient, energy efficient control scheme for a CPS from all the schemes considered. We also identify the resiliency limitations of our scheme to different jamming strategies.

The use of wireless communication enables the creation of CPSs by adding low cost and non-intrusive monitoring and control systems to legacy and new critical infrastructure like WDNs. Currently, WDN pumps and valves are actuated manually to meet the daily fluctuation of users demands. Next generation WDNs are being deployed with automatic controllers. The goal is to eventually integrate sensing with actuation to make the WDNs more responsive to user demand, and more efficient in their use of water and pump energy.

CPSs enable the actuation of physical processes by processing sensed data with control schemes derived from control theory. The use of a control theoretic approach provides safety and performance guarantees [2]. However, the combination of wireless communication and control comes with risks. If the wireless communication is disrupted, the control schemes will not have access to temporally relevant data, and performance and safety guarantees can not be met. In this paper we focus on the secure design approach of CPSs considering two types of control schemes, time-triggered and event-triggered.

Time-Triggered Control (TTC) schemes [3] are commonly used because they are easy to analyse and design, and they provide safety and performance guarantees as long as they receive sensor data at a constant, periodic rate. They are difficult to use on CPSs based on low-cost wireless sensors and actuators. These devices have a constrained energy capacity which is quickly drained by constant periodic use [4].

Event-Triggered Control (ETC) schemes [5] are a solution to the high communication costs of periodic time-triggered transmissions. ETC schemes only transmit new measurements and control signals when necessary while guaranteeing stability or performance requirements. This enables a creation of more efficient CPSs that are easier to deploy and where the need for frequent battery changes is mitigated.

TTC and ETC schemes are unable to maintain guarantees if they cannot receive new measurements in a timely fashion. CPSs using wireless communication are susceptible to failure if their communication is disrupted by a security attack [6]. Many examples can be given such as the Stuxnet attack on Iranian uranium processing facilities [7], the multistage attack on SCADA system in the Ukrainian power grid [8], and the denial-of-service attack on a power grid in Germany [9]. There is currently no design approach to address the performance and safety guarantees of CPSs when attacked by cyber attacks.

We follow a design approach that recognises that CPS communication and control system are tightly coupled. This is a system level design problem, as an attack on the communication can prevent up-to-date information from reaching the controller. This will prevent the stable operation of the system. This specific problem is referred to as the communication and control systems co-design problem [10], [11].

Our approach is to evaluate the resilience of two types of controllers, TTC and ETC, to various plausible jamming

attacks [12]. We then implement these controllers on a CPS test-bed called the WaterBox [13]. For the evaluation we use a real jammer, with various levels of jamming sophistication, to do real disruption of our CPS test-bed. We use the results of the disruption as an indication of the success of our design.

The contributions of this work are threefold:

- We present and demonstrate the use of a design approach to choose a controller for CPS. Our design approach identifies the most jamming resilient, resource efficient control scheme of all the schemes considered. We increase the resilience of our controller with a dynamical estimator, and identify the resiliency limitations of our scheme.
- We present several jamming strategies of different levels of sophistication. These are developed in order to measure the resilience of our ETC-based controller.
- To the best of our knowledge, this paper provides the first real test-bed evaluation of the resilience of periodic and ETC schemes to different kinds of jamming attacks aimed at communication.

The rest of the paper is organized as follows: Sec. 2 surveys related work. Sec. 3 discusses the control and communication system model. Sec. 4 presents the design of the three jamming strategies. We present the evaluation results in Sec. 5 and end the paper in Sec. 6 with brief concluding remarks.

II. Background and Related Work

Traditional periodic controllers increase the cost of sensors/actuators and their maintenance requirements which makes them unfit for the use in CPSs [4]. As an alternative, ETC schemes have been studied widely by the control community (e.g. [5], [14], [15]). Their use has been further enhanced by integration with communication protocols [16]–[18].

The majority of the literature on ETC schemes integrated with communication protocols have not considered the affects of cyber-physical security attacks. This is an oversight if they are going to be used to monitor or control critical infrastructure systems like WDNs. Notable exceptions are [19]–[22]. The method in [19] provides resilience only to a periodic jamming attack. This is an unrealistic assumption, as jammers can be more sophisticated rather that only emit periodically. This has been complemented in [20]–[22] where the jamming strategy is neither known nor pre-fixed. However, the system is only jammed for a certain percentage of time on average to allow for sufficiently regular data flow.

Contrarily, we identify where this resiliency limitation of various control schemes lies for three jamming strategies. Our protocol aware jammer is more sophisticated than those used in above mentioned studies on ETC security. Similar jamming strategy is given in [23]. However, unlike the design in [23], we ensure the adaptivity property. This means that our jammer can adapt to the change in the communication pattern (i.e. the change in the underlying physical phenomenon). We provide an experimental implementation which limits us from making unrealistic assumptions such as in [22] that a node can accurately differentiate a packet lost to a jammer to one lost due to network link quality issues.

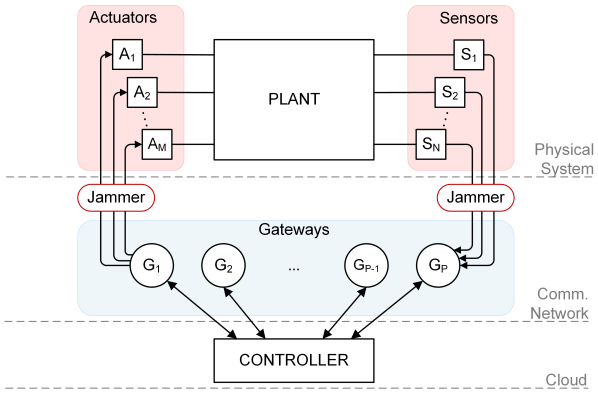

Fig. 1. The CPS architecture with a jammer affecting both, uplink and downlink communication channels.

Next, we present the system model for which we evaluate the resilience against different jamming strategies.

III. System Model

In this section we begin our design approach by describing the dynamical control system and the communication models. We use these to represent a smart water network which is emulated on the WaterBox test-bed [13] in Sec. V. In Fig. 1 we present a diagram of the CPS considered in this paper. The CPS consists of a large complex physical process (the plant) and the control and management system which are connected via a wireless communication network. In the case of the WaterBox, the communication channel is a WiFi network.

We assume that sensors and actuators are collocated on sensor/actuator nodes denoted by $j = 1, \ldots, N$. N is the total number of nodes that share the same frequency channel. Nodes communicate with the controller in a single-hop fashion. The communication from sensor/actuator nodes to the controller and vice versa can be attacked by jamming signals.

A. Control Model

We consider a linear time-invariant dynamical system:

$$\dot{\xi}(t) = A\xi(t) + Bv(t), \quad \xi(0) = \xi_0 \tag{1}$$

where $\xi(t) \in \mathrm{R}^n$ denotes the state input and $v(t) \in \mathrm{R}^m$ denotes the control input at time t. The matrix $A \in \mathrm{R}^{n\times n}$ is the state matrix and $B \in \mathrm{R}^{n\times m}$ is the input matrix.

In a conventional time-triggered control (TTC), the sampling instants t_k, $k \in \mathrm{N}$ occur periodically, i.e. $t_k = kh$ for $h > 0$, $k \in \mathrm{N}$. Control actions are also sent periodically and they work in a sample-and-hold fashion:

$$v(t) = K\xi(t_k), \quad \forall t \in [t_k, t_{k+1}]. \tag{2}$$

The feedback gain matrix is denoted as $K \in \mathrm{R}^{n\times n}$. At the start of each period $t_k = kh$, sensor/actuator nodes send their state to a *centralised TTC controller*. The controller generates a control input based on the received sensor/actuator states, then it sends the control input back to the nodes for actuation.

In event-triggered control (ETC), the controller only updates and sends a control input if pre-defined triggering condition is satisfied at $t_k = kh$ for $h > 0$, $k \in \mathrm{N}$. We refer to this as to an event and it can be expressed in terms of the measurement error, $\epsilon(t_k) = |\xi(t_k) - \xi(t)| \leq \eta$ for $t \in [t_k, t_{k+1}]$ that exceeds the predefined threshold value η. The ETC approach differs from the TTC where events are transmitted regularly regardless of the state of the plant. ETC schemes only transmit an event if one actually occurs which makes them more energy efficient when compared to TTC. In the case of ETC, we consider both centralised and decentralised controllers.

The control input $\upsilon(t_k)$ at time t_k of the *centralised ETC controller* is given by

$$\upsilon(t_k) = \begin{cases} K\xi(t_k), & \text{if} \quad \theta_k = 1 \\ K\xi(t_{k-1}), & \text{if} \quad \theta_k = 0 \end{cases} . \tag{3}$$

The indicator function $\theta_k = 1$ indicates that the triggering condition is satisfied, $\theta_k = 0$ indicates that the triggering condition is not satisfied at time t_k. The controller needs to receive states from each sensor/actuator nodes periodically in order to evaluate the triggering condition. Based on the outcome, it sends or not an updated control action to the sensor/actuator nodes.

In the case of the *decentralised ETC controller*, the triggering condition is distributed to each sensor/actuator node j. The nodes only send their data to the controller when the threshold is violated. The controller can recalculate the control action using *synchronous approach* when upon receiving the state of at least one node, it requests state from other nodes. Based on these, it generates and sends a new control action:

$$\upsilon(t_k) = \begin{cases} K\xi(t_k), & \text{if } \exists j \in \{1, \ldots, N\} : \theta_{k_j} = 1 \\ K\xi(t_{k-1}), & \text{if } \forall j \in \{1, \ldots, N\} : \theta_{k_j} = 0 \end{cases} . \tag{4}$$

$\theta_{k_j} = 1$ indicates that the triggering condition is satisfied at node j. The decentralised ETC controller can also use *asynchronous approach* where only the corresponding sensor/actuator node's state (i.e. node that reported the threshold violation) is used to update the control action, i.e. if $j \in \{1, \ldots, N\}$ and $\theta_{k_j} = 1$, then $\xi(t_k) = [\xi_1(t_{k-1}), \xi_2(t_{k-1}), \ldots, \xi_j(t_k), \ldots, \xi_N(t_{k-1})]^T$.

Next we describe a communication network that is used to pass the information between sensor/actuator nodes and the controller. We provide three communication modes to support different types of controllers presented here.

B. Communication Model

To support the information exchange between the system and the controller, a TDMA-based protocol [17] is used. It allows several sensor/actuator nodes to share the same frequency channel. Time is divided into intervals T_i, $i = 1, \ldots, \infty$, of length T seconds. Each T_i is divided into a number of sub-slots allocated to nodes. The node's j slots repeat every T seconds. The structure and purpose of the sub-slots used for communication in different control schemes are briefly explained next with the graphical representation in Fig. 2. Note how the communication modes descriptions mirror the descriptions of the control schemes that use them.

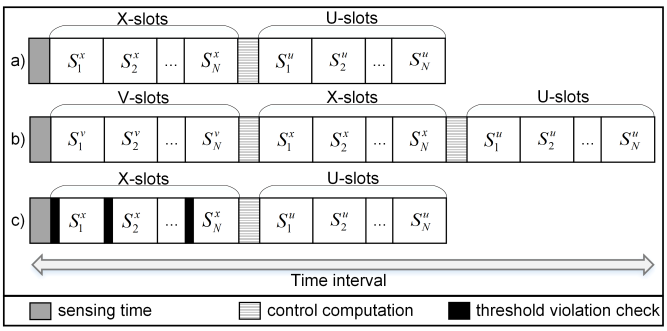

Fig. 2. The structure of a TDMA-based time interval for: a) TTC and PETC scheme b) PSDETC scheme c) PADETC scheme.

- *Periodic centralised time-triggered (TTC) & event-triggered control (PETC)* - An interval T_i is divided into two subsets of time slots, S_j^x and S_j^u, which are separated by a time delay d_c. Each sensor node transmits its measured state to the controller during the X-slot S_j^x. The controller returns the control action to the sensor/actuator nodes during the U-slot S_j^u. The delay d_c allows controller to compute the control action.
- *Periodic synchronous decentralised ETC (PSDETC)* - This mode adds an additional subset of violation slots, V-slots S_j^v. V-slots are used by sensor/actuator nodes to report the threshold violation to the controller. Consequently, the controller sends request to other nodes during the X-slot. These nodes transmit their measurements during the X-slot also which is followed by a new control action during the U-slot.
- *Periodic asynchronous decentralised ETC (PADETC)* - Compared to PSDETC, V-slots are merged with X-slots. If there is the threshold violation, the node transmits the new measurement; otherwise, it skips the communication until next time interval. The controller computes the control input based on the received information only and returns it to all sensors in the U-slots.

The length of different subsets depends on the application (i.e. underlying physical phenomenon) as well as on the complexity of the control infrastructure.

C. Estimator Model

If the communication channel becomes unreliable, the controller may not receive up-to-date states or notification of a triggering condition being met. This would cause the associated system's performance degradation or in a more severe case a complete failure. In order to avoid system failure in the case of communication failure, there is a need for the estimation of corrupted or lost data. The topic of estimation over lossy networks is widely discussed in control systems literature [2]. In this paper, estimates are computed recursively using a time-varying Kalman filter [24].

We assume that the system in Eq. 1 is completely observable, and each sensor/actuator node can access only one of the system's states. First, the initial state estimate is set to zero (i.e. $\hat{\xi}_i(0) = 0$). The state estimation error at the time

t, represents the difference between the actual state $\xi(t)$ and the estimated state $\hat{\xi}(t)$, i.e. $e(t) = \xi(t) - \hat{\xi}(t)$. The corrected version of the state estimate $\hat{\xi}(t)$ is given by

$$\hat{\xi}_c(t) = \hat{\xi}(t) + F_k e(t) \tag{5}$$

where F_k is the Kalman filter gain. The Kalman filter gain is chosen to minimize the estimation error covariance matrix $P(t) = E[e(t)e(t)^T]$ [24]. Finally, the predicted state estimate at the time $t+1$ is defined as

$$\hat{\xi}(t+1) = A\hat{\xi}_c(t) + B\upsilon(t). \tag{6}$$

If there is no actual state coming (e.g. due to the jamming activity), the estimator predicts $\hat{\xi}(t+1)$ based on its previous estimate. Implementation details are given in Sec. V-A.

All presented control schemes share the same vulnerability, which is their use of wireless communication for sensed data and control actions. For the next part of our design approach we examine the vulnerability in more detail. We present three strategies that can be used to jam the wireless communication used by a CPS, prevent up-to-date data from reaching the controller, and render the underlying physical system unstable. It is important for us to understand the way that the attacker functions so that it can be replicated and used to evaluate the resilience of our control schemes.

IV. Jammer Model and Attacking Strategies

The next part of our design approach is to understand the potential threats to the CPS's wireless communication. We present three plausible jamming strategies in order to evaluate the resilience of our control schemes to communication loss. The strategies include, constant, random, and protocol aware jamming. We evaluate the strategies based on the ease of detection and their effectiveness. The plausibility of the jamming strategies is demonstrated through implementation.

A. Jamming Strategies

In this paper we assume that the jammer aims to disrupt the wireless communication on the measurements channel (i.e. from sensor/actuator nodes to the controller) as shown in Fig. 1. The considered jamming strategies which vary in terms of undetectability and effectiveness of jamming as per Fig. 3 are described below.

Constant jammer - Transmits all the time. It is guaranteed to be completely effective if the jammer is well placed and has sufficient transmission power. Every message in the jammer's range will be disrupted. The negative part of constant jamming is that it is easy to locate.

Random jammer - Transmits for fixed periods of time. We maintained a fixed transmission period of 500ms, and varied the sleep periods between 200, 300, 400, 500, 600, 700, and 800ms with a uniform random distribution.

Protocol aware jammer - The aim of this method is to jam all of the messages by learning both the transmission period and phase of the target node. It has the shortest transmission period, of 250ms, making it the least detectable.

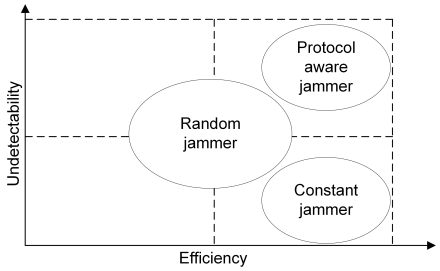

Fig. 3. Effectiveness and undetectability of the proposed jamming models.

B. Jamming Strategy Metrics

We describe our jamming strategies based on two metrics, the effectiveness of the strategy at disrupting the CPS and the difficulty of detecting the location of the jammer.

The *effectiveness of the attack* assumes that we want to completely disrupt our target CPS. To do this we measure the packet delivery ratio (i.e. a count of the number of messages received by the controller over the messages sent by the target sensor/actuator node), and the deviation of the controlled system from safe steady state operation. In the case of our WaterBox test-bed, this is the deviation of water level in tanks.

Difficulty of detection follows from our stated goal of complete system disruption. Once the system is disrupted, it's owners will notice, and come looking for the jammer. The jammer will be easy to locate by radio triangulation. The less time the jammer spends in transmission, the longer it will take to be located and recover the correct operation of the CPS.

We do not use power saving as a metric, due to the fact that a successful jammer could possibly be found and terminated before its batteries fail, but can not ignore it as a positive side effect for jammers that spend less time in transmission.

In the next section we present the final part of our design approach, the evaluation test-bed. The test-bed allowed us to evaluate the resilience of the control schemes to jamming strategies with real radio and physical effects.

V. Evaluation

A. WaterBox Evaluation Platform

In the previous two sections we presented the control schemes under consideration, and the jamming strategies that can disrupt the flow of data to our controllers, and render the system unstable. In this section, we present the WaterBox test-bed (see Fig. 4) that we use to evaluate the resilience of various control schemes to various jamming strategies.

The WaterBox is designed to emulate a WDN. It allows us to experiment with different control schemes that guarantee the system's operation within safe bounds (i.e. between the maximum and the minimum allowed water level in the tanks). Each sensor/actuator node is connected to a local WiFi network to send the measured water level to the controller. The controller's goal is to provide control actions that ensure system stability and performance guarantees.

The WaterBox consists of lower tank, upper tank and three small District Meter Area (DMA) tanks. The lower tank pump

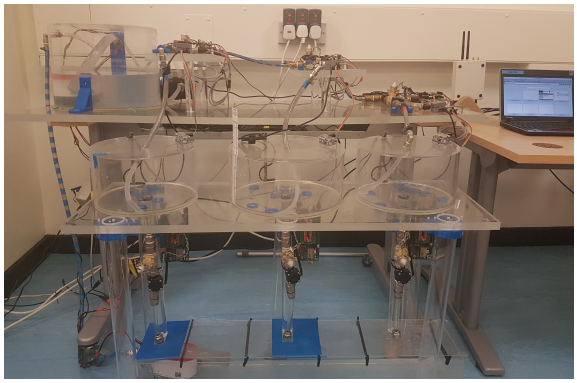

Fig. 4. The WaterBox test-bed.

supplies water to the upper tank which is then pumped to the three small DMA tanks by the assistant pump and/or the powerful pump. The water flow is controlled by motorized gate valves which are based on an Edison development board. Each DMA tank is equipped with one set of in/out-valves. Each valve is fit with a sensor/actuator node. The WiFi transmission levels of nodes are set to their default value of 31dBm.

The system can work in two different modes for which the state matrix A is given as a zero matrix, i.e. $A_w = A_p = O_{3\times 3}$, and the input matrix B is given as

$$B_w = 10^{-5} \times \begin{bmatrix} 0.1436 & -0.0170 & -0.0164 \\ -0.0098 & 0.1060 & -0.0100 \\ -0.0139 & -0.0139 & 0.1492 \end{bmatrix},$$

$$B_p = 10^{-5} \times \begin{bmatrix} 0.7666 & -0.0493 & -0.0457 \\ -0.0274 & 0.5848 & -0.0279 \\ -0.0393 & -0.0432 & 0.1492 \end{bmatrix}.$$

The control law in Eq. 2 works in a sample-and-hold fashion with the feedback gain matrix K given as

$$K_w = \begin{bmatrix} 99950 & 3029 & 872 \\ -3014 & 99940 & -1679 \\ -922 & 1652 & 99982 \end{bmatrix},$$

$$K_p = \begin{bmatrix} 9998.5 & 167.1 & 41.0 \\ -166.6 & 9997.9 & -116.0 \\ -43.0 & 115.3 & 9999.2 \end{bmatrix}.$$

The *'weak mode'* is represented by (A_w, B_w, K_w) and it simulates low water demand that a water network would experience during the night when only the assistant pump is on. The *'powerful mode'* is represented by (A_p, B_p, K_p) and it simulates high water demand during the day when the powerful pump is enabled.

The normal operation of the WaterBox, regardless of the control scheme, starts in powerful mode with both pumps on. The switching from powerful to weak mode happens when $|S(-K_p\xi(t) + \alpha_p^{in})|_1 < 180°$. $S(\cdot)$ is a function that maps actuator saturation and quantization to represent the degree to which valves are open. The valves themselves are discrete, and open and close in steps of $10°$. The term $|\cdot|_1$ is L^1-norm, or sum of the entries of the resulting vector, and α_p^{in} denotes the degree that the in-valves are open at equilibrium.

To facilitate the need for switching back to powerful mode and therefore the need for the communication, we keep the out-valves fully open. The water level will drop and when the minimum water level h_{l_j} is reached (i.e $\xi_j(t) \leq h_{l_j}$), the system switches back to the powerful mode.

TABLE I
WATERBOX EVALUATION SETUP

	Parameter	Value	Description
Timing Param.	T	1s	Interval duration
	S_j	20ms	Sensing slot size
	S_j^x, S_j^u, S_j^v	50ms	X-slot, U-slot, V-slot size
	d_c	10ms	Computing control delay
	d_g	5ms	Threshold violation check delay
Control Schemes Setup	h_{d_j}	3cm	Maximum water level for node j
	h_{l_j}	1cm	Minimum water level for node j
	σ_1	0.04	PETC violation parameter
	σ_2	0.05	PSDETC violation parameter
	μ, ρ	85, 0.95	PADETC violation parameters

The control schemes tuning parameters, as well as the communication timing parameters are presented in Table I. For more details on the model, we refer the reader to [17].

B. Jamming Evaluation Platform

An experimental radio jammer platform is created to enable us to study the effects of communication disruption on various control schemes. The platform consists of a separate WiFi Sniffer and Jammer, controlled through a laptop (ASUS X555U running Ubuntu 16.04). The platform aims to jam all three of the in-valve sensor/actuator nodes on the WaterBox.

The Sniffer laptop is positioned at a distance of 185cm, 138cm, and 110cm from the three respective in-valve sensor/actuator nodes on the WaterBox (DMA0, DMA1, DMA2). We use the Tshark packet sniffing tool to put the RTl8821ae WiFi card into promiscuous mode in order to capture all of the WiFi traffic in the range of the laptop.

Jammer is located at a distance of 170cm, 120cm, and 95cm from the in-valve sensor/actuator nodes. The jamming signal is pure white noise (uniform random) generated and transmitted by a USRP B210 SDR at its maximum transmit power of 10dBm. The Jammer is connected by USB 3.0 to the Sniffer, and powered by an external power supply. We control the Jammer through GNU radio (version 3.7.9).

The Sniffer and Jammer are connected together in a Python 2.7 script. The script makes calls to Tshark that gathers data, processes that data, and then uses GNU radio libraries to actuate the jamming strategies.

In the Alg. 1 we present the implementation of the WiFi protocol aware jamming strategy for periodic CPS nodes that uses the following functions:

- *calcPhase(·)* - In order to synchronize the jammer and target device, the Tshark listener and jammer are called concurrently. The timings of both functions are compared to determine if a phase offset exists and if so, is applied to the jammer trigger in future time slots.
- *triggerJammer(·)* - This function triggers the SDR (jammer) for period t_d and with phase offset P_h.
- *updatePeriod(·)* and *triggerJammer(·)* are run concurrently to determine if there is a phase offset between

Algorithm 1: WiFi Protocol Aware Jamming Strategy

Input : Listening Duration t_l, Jamming Duration t_d
$t_0 = time.now(\cdot)$ // Records epoch time
$\alpha \leftarrow triggerTShark(t_l)$ // Calls Sniffer in monitor mode
$\omega \leftarrow filterMsg(node, router, data_length)$
// Reduces stored set α to target node(s)
for $key \in \omega[1:]$ **do**
  $\beta \leftarrow (key - t_0)$
  $t_0 \leftarrow key$
end
// Calculates array message inter arrival time β
$P \leftarrow median(\beta)$
// Median of β used to determine periodicity of node
$P_h \leftarrow calcPhase(\cdot)$
// Calculates phase offset to synchronize
// Jammer with target device to be jammed
while $True$ **do**
  $triggerJammer(P, t_d, P_h)$ // Calls jamming function
  $P, P_h \leftarrow updatePeriod(\cdot)$ // Checks for changes to P, P_h
end

the target node and the jammer. If so this is passed to recalculate the new trigger time. This ensures adaptivity when the underlying physical phenomenon changes.

C. Experimental Results

This section summarizes the results of our design approach evaluation. These allows us to understand the resilience of our controllers, and the affects of our attackers on a WDN CPS.

The resilience of control schemes and the effectiveness of the attack are evaluated based on the following metrics:

- Deviation of the water level from the steady-state value (in %, maximum value of three DMA tanks) - It indicates the maximum system's deviation from the safe operating conditions which is critical for water networks.
- Packet Delivery Ratio (PDR) (in %, the average of three DMA tanks) - It is the ratio of the number of messages received by the controller and the number of messages sent by the target node.
- Difficulty of detection (in %) - It represents the percentage of time jammer spends on transmitting the signal.

First we give the result that represents the normal system operation for which the communication is unaffected by jammers. This case is considered as the baseline for the evaluation of four WaterBox control schemes (TTC, PETC, PSDETC and PADETC) under constant, random and protocol aware jamming attacks. The number of experiments was selected experimentally by analysing the variance of the results (i.e. under 2% of the mean value). Experiment time is 600s, unless the system experiences failure earlier due to jamming activity.

1) Baseline Performance: During the normal operating conditions, the WaterBox provides identical responses under different control schemes (the control goal and the operating conditions are the same). Therefore, we present the system's response only for the TTC scheme (see Fig. 5).

It takes 33 seconds (on average) for the system to reach the steady-state value and continues operating within the safe

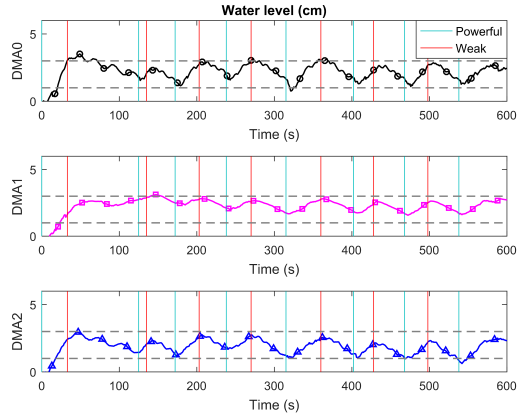

Fig. 5. System's response under normal operating conditions for TTC.

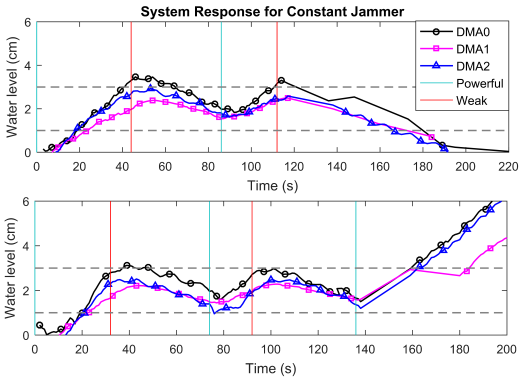

Fig. 6. System's response under constant jamming for TTC: 1) Top - Jamming started while system was in the 2nd weak mode 2) Bottom - Jamming started while system was in the 3rd powerful mode.

bounds (in our case between 1cm and 3cm). Due to the nature of the system (all three tanks are controlled by a single pump) and the limited size of the DMA tanks, the system may experience overshoots and undershoots. This is expressed through the deviation metric in Table II. As the behaviour of our system is known and bounded, the acceptable deviation is up to 15%. Everything over is considered as the failure.

The system achieves very high PDR values for all control schemes. This is due to the fact that the communication is unaffected and there is no jamming activity present. The results for different control schemes are presented in Table II.

2) Constant Jamming: The constant jamming transmissions are completely effective in terms of achieved PDR and the system's deviation from the normal operation conditions. The communication is completely blocked and the system experiences a complete failure for all control schemes, i.e. the system overflows or underflows (see Table II). We present the system's response only for TTC scheme as other control schemes perform similarly (see Fig. 6).

The negative side of this jamming strategy is the ease of detection. As the constant jammer transmits 100% of the time it can be easily located and stopped.

3) Random Jamming: The jammer chooses its transmissions randomly which results in lack of the guarantees for system failing. As it can be seen in Fig. 7, the random

TABLE II
THE EVALUATION OF THE RESILIENCE OF VARIOUS CONTROL SCHEMES TOWARDS CONSTANT, RANDOM AND PROTOCOL AWARE JAMMING.

Control scheme	**Baseline Jamming** 0% jamming time		**Constant Jamming** 100% jamming time		**Random Jamming** 50% jamming time		**Protocol aware** 25% jamming time	
	Deviation (%)	PDR (%)	Deviation (%)	PDR (%)	Deviation (%)	PDR (%)	Deviation (%)	PDR (%)
TTC	12.42	99.13	FAIL*	6.26	61.84	78.15	FAIL*	21.90
PETC	10.11	99.50	FAIL*	5.88	15.33	81.46	FAIL*	24.83
PSDETC	11.39	98.81	FAIL*	6.35	FAIL*	64.67	FAIL*	50.82
PADETC	12.98	98.07	FAIL*	6.87	FAIL*	59.99	**36.33**	**88.31**

*the system overflows or underflows as the system deviation is larger than $\pm 15\%$

TABLE III
PROBABILITY OF RANDOM JAMMING WRT THE JAMMING DURATION

Duration(s)	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Probab.(%)	20	30	40	50	60	70	80	90	100

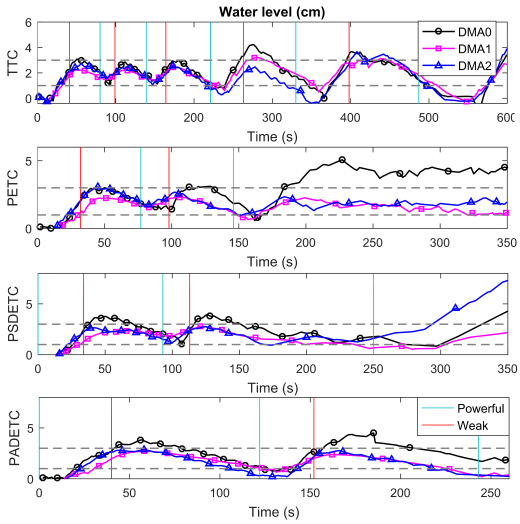

Fig. 7. System's response under random jamming for different control schemes. Jamming started while system was in the 4th weak mode for TTC, 3rd powerful mode for PETC, 2nd weak mode for PSDETC and PADETC.

jammer leads to a complete failure for PSDETC and PADETC control schemes, while TTC and PETC continue to operate, but outside the acceptable deviation range (see Table II).

While this jamming strategy is more difficult to detect compared to constant jamming, there is a big variance in the results and no guarantees can be provided. Table III shows the probability of jamming the system with respect to the jamming duration. As it can be seen, with a random jamming period of 500ms, there is high probability of 60% to jam the system. Note that the results presented in Table II for the random jammer are not averaged for all experiments due to the lack of consistency (i.e. the high variance of the results).

4) Protocol Aware Jamming: The protocol aware jammer transmits only 25% of the time. Compared to the constant and the random jammer, it is the least detectable. By learning the transmission period and phase, it is able to completely jam TTC and PETC schemes. In both cases the PDR drops under 25% and the system fails (see Table II). The results are depicted in Fig. 8. For all of the experiments the jammer was

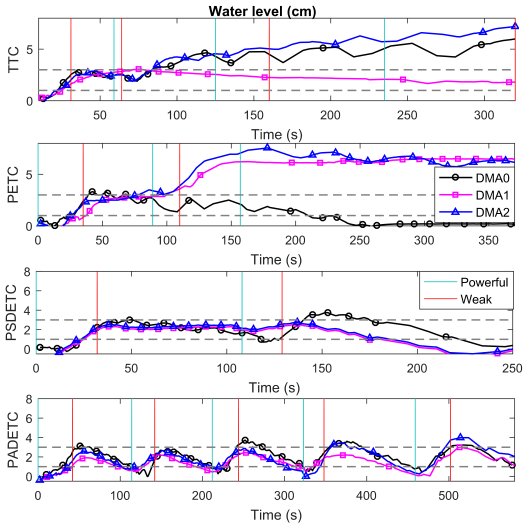

Fig. 8. System's response under protocol aware jamming for different control schemes.

started once the system reached the steady-state.

The reduction in the transmissions for the PSDETC scheme increases its resilience towards the protocol aware jammer. This can be observed via the PDR that is doubled compared to TTC and PETC. However, as the time evolves the lack of information affects the system's stability. The only scheme that shows the resilience to our protocol aware jammer is the PADETC. It achieves high PDR and the system is able to maintain the normal operation while under jamming attack. As it can be seen from Fig. 8, the deviation from the normal operating conditions is outside the safe bounds by 21.33%. This is contributed to the lost data which delays the controller's operation (and therefore the actuation). The system reacts slower than expected. We overcome this problem by extending the current model by the dynamical estimator.

5) PADETC with the Estimator under Protocol aware Jamming: The resilience of the PADETC is further improved through the addition of the dynamical estimator. The data that is lost due to jamming activity is estimated as presented in Sec. III-C. The data estimation is performed every 2 seconds if no data has been received during that period.

The deviation from the normal operating conditions is reduced from 36.33% to 10% (on average). The control decisions are made as soon as the threshold is violated. The results are depicted in Fig. 9. For more clear results we present

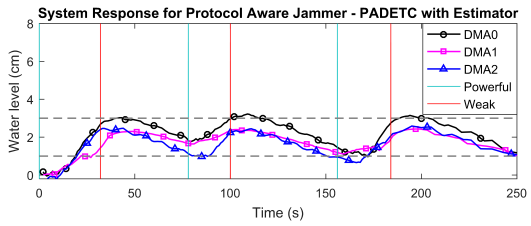

Fig. 9. System's response under protocol aware jamming for PADETC with a dynamic estimator.

the system's response for the first 250s; however, the system's behaviour repeats as the time evolves. Note that the PADETC-based scheme for CPSs that features an estimator can be used to increase the resilience of the system to communication disruption in general, not just from jamming attacks.

D. Design Approach Results

To summarise the results of our design approach, we can see that the PADETC control strategy with an estimator is resilient to protocol aware jamming. It is also the most energy efficient as the sensors transmit states asynchronously to the controller when the triggering condition is satisfied.

Our design approach also gives us an understanding of the resiliency limits of PADETC to different jamming strategies. No controller can cope with the constant jamming strategy. This is a limit that no control strategy alone can overcome. This is made less severe by realising that a constant jammer is easy to detect, and can be found and stopped with little effort. The resilience of PADETC to the random jammer is highly variable. When the jamming window is more than 50%, the scheme will fail. This is another limit, but a probabilistic one. As the jamming window of the random jammer increases, its detectability also increases.

VI. Conclusions

In this paper we present a design approach to create a CPS that is secure to communication jamming. Our approach achieves this goal through the identification of the most resilient control scheme to communication jamming attacks.

We demonstrate our approach by evaluating the TTC and ETC control schemes. We created plausible threats in the form of jammers using different strategies. We evaluated the resilience of the controllers to message loss caused by various jamming strategies. This method allowed us to witness and measure the success of each attack on a real CPS test-bed. The results of our approach clearly showed that the PADETC control scheme is resilient to protocol aware jamming. Results also indicated the resilience limitations of this scheme.

Our work is important because it adds security to the communication and control co-design problem. We increase the fidelity of our results by evaluating on a real test-bed so that we can measure the impact of the attacks. This is especially topical because service providers like water and electricity utility companies are currently studying the use of CPSs. We have clearly demonstrated a method to assess the security weaknesses of certain types of control schemes, and shown that certain control schemes are more robust to communication attacks than others.

References

[1] R. Rajkumar, I. Lee, L. Sha, and J. Stankovic, "Cyber-physical systems: The next computing revolution," *Proc. of the 47th Design Autom. Conf.*, pp. 731–736, 2010.

[2] J. P. Hespanha, P. Naghshtabrizi, and Y. Xu, "A survey of recent results in networked control systems," *Proc. of the IEEE*, vol. 95, no. 1, pp. 138–162, 2007.

[3] K. J. Åström and B. Wittenmark, *Computer-controlled Systems (3rd Ed.)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1997.

[4] M. Mazo and P. Tabuada, "Decentralized event-triggered control over wireless sensor/actuator networks," *IEEE Trans. on Autom. Control*, vol. 56, no. 10, pp. 2456–2461, 2011.

[5] W. P. M. H. Heemels, M. C. F. Donkers, and A. R. Teel, "Periodic event-triggered control for linear systems," *IEEE Trans. on Autom. Control*, vol. 58, no. 4, pp. 847–861, 2013.

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

[7] R. Langner, "Stuxnet: Dissecting a cyberwarfare weapon," *IEEE Security Privacy*, vol. 9, no. 3, pp. 49–51, 2011.

[8] E-ISAC, "Analysis of the cyber attack on the ukrainian power grid," *Defense Use Case*, 2016.

[9] J. Goldman. (2012) European power grid hit by cyber attack. [Online]. Available: https://www.esecurityplanet.com/network-security/european-power-grid-hit-by-cyber-attack.html

[10] P. Park, S. C. Ergen, C. Fischione, C. Lu, and K. H. Johansson, "Wireless network design for control systems: A survey," *IEEE Communications Surveys Tutorials*, vol. PP, no. 99, pp. 1–1, 2017.

[11] B. Wu, M. D. Lemmon, and H. Lin, "Formal methods for stability analysis of networked control systems with IEEE 802.15.4 protocol," *IEEE Trans. on Control Sys. Tech.*, vol. PP, no. 99, pp. 1–11, 2017.

[12] W. Xu, W. Trappe, Y. Zhang, and T. Wood, "The feasibility of launching and detecting jamming attacks in wireless networks," in *Proc. of the 6th ACM Int. Symp. on Mobile Ad Hoc Netw. and Comput.*, 2005, pp. 46–57.

[13] S. Kartakis, E. Abraham, and J. A. McCann, "Waterbox: A testbed for monitoring and controlling smart water networks," *1st ACM Int. Workshop on Cyber-Phys. Sys. for Smart Water Netw.*, pp. 8:1–8:6, 2015.

[14] M. Mazo and P. Tabuada, "On event-triggered and self-triggered control over sensor/actuator networks," *47th IEEE Conf. on Decision and Control*, pp. 435–440, 2008.

[15] X. Wang and M. D. Lemmon, "Event-triggering in distributed networked control systems," *IEEE Trans. on Autom. Control*, vol. 56, no. 3, pp. 586–601, 2011.

[16] R. Blind and F. Allgwer, "Analysis of networked event-based control with a shared communication medium: Part I - Pure ALOHA," *IFAC Proc. Volumes*, vol. 44, no. 1, pp. 10 092–10 097, 2011.

[17] S. Kartakis, A. Fu, M. Mazo, and J. A. McCann, "Communication schemes for centralized and decentralized event-triggered control systems," *IEEE Trans. on Control Syst. Technol.*, pp. 1–14, 2017.

[18] M. Vilgelm, M. H. Mamduhi, W. Kellerer, and S. Hirche, "Adaptive decentralized MAC for event-triggered networked control systems," *Proc. of the 19th Int. Conf. on Hybrid Sys.: Comp. and Control*, pp. 165–174, 2016.

[19] H. S. Foroush and S. Martnez, "On event-triggered control of linear systems under periodic denial-of-service jamming attacks," *51st IEEE Conf. on Decision and Control*, pp. 2551–2556, 2012.

[20] C. D. Persis and P. Tesi, "Resilient control under denial-of-service," *IFAC Proc. Volumes*, vol. 47, no. 3, pp. 134–139, 2014.

[21] A. Cetinkaya, H. Ishii, and T. Hayakawa, "Event-triggered control over unreliable networks subject to jamming attacks," *54th IEEE Conf. on Decision and Control*, pp. 4818–4823, 2015.

[22] V. S. Dolk, P. Tesi, C. D. Persis, and W. P. M. H. Heemels, "Event-triggered control systems under denial-of-service attacks," *IEEE Trans. on Control of Netw. Sys.*, vol. 4, no. 1, pp. 93–105, 2017.

[23] Y. W. Law *et al.*, "Energy-efficient link-layer jamming attacks against wireless sensor network MAC protocols," *Proc. of the 3rd ACM Workshop on Security of Ad Hoc and Sensor Netw.*, pp. 76–88, 2005.

[24] T. Kailath, B. Hassibi, and A. H. Sayed, *Linear estimation*. Upper Saddle River, NJ : Prentice-Hall, 2000.