# Minimax Control For Cyber-Physical Systems under Network Packet Scheduling Attacks[*]

Yasser Shoukry
Electrical Engineering
UC Los Angeles
yshoukry@ee.ucla.edu

Jose Araujo
ACCESS Linnaeus Center
KTH Royal Institute of Technology
araujo@kth.se

Paulo Tabuada
UC Los Angeles
tabuada@ee.ucla.edu

Mani Srivastava
UC Los Angeles
mbs@ee.ucla.edu

Karl H. Johansson
ACCESS Linnaeus Center
KTH Royal Institute of Technology
kallej@kth.se

## ABSTRACT

The control of physical systems is increasingly being done by resorting to networks to transmit information from sensors to controllers and from controllers to actuators. Unfortunately, this reliance on networks also brings new security vulnerabilities for control systems. We study the extent to which an adversary can attack a physical system by tampering with the temporal characteristics of the network, leading to time-varying delays and more importantly by changing the order in which packets are delivered. We show that such attack can destabilize a system if the controller was not designed to be robust with respect to an adversarial scheduling of messages. Although one can always store delayed messages in a buffer so as to present them to the control algorithm in the order they were sent and with a constant delay, such design is overly conservative. Instead, we design a controller that makes the best possible use of the received packets in a minimax sense. The proposed design has the same worst case performance as a controller based on a buffer but has better performance whenever there is no attack or the attacker does not play the optimal attack strategy.

[*]This work is supported by the Knut and Alice Wallenberg Foundation, the Swedish Research Council, the HYCON2 EU project, the NSF grant CNS-1136174, and by DARPA under agreement number FA8750-12-2-0247. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA or the U.S. Government

*HiCoNS'13,* April 9–11, 2013, Philadelphia, Pennsylvania, USA.
Copyright 2013 ACM 978-1-4503-1961-4/13/04 ...$15.00.

## Categories and Subject Descriptors

J.2 [**Computer Applications**]: PHYSICAL SCIENCES AND ENGINEERING Engineering, Electronics

## Keywords

Out-of-order packets; Cyber-physical systems (CPS); networked control systems (NCS); Security; Resilient control; Minimax control; Out-of-order messages; Long delays; Estimation

## 1. INTRODUCTION

The increased coupling between embedded computing technologies and modern control systems has opened the door for developing many engineering systems with growing complexity. In such systems, commonly termed *cyber-physical systems* (CPS), information from the physical world is quantized and processed using digital electronic components, and decisions taken by these "cyber components" are then applied to the physical world [20, 22]. Unfortunately, this tight coupling between cyber components and the physical world oftentimes leads to systems where increased sophistication comes at the expense of increased vulnerability and security weaknesses. There exist several examples of attacks on CPSs such as the first-ever control system malware called Stuxnet [26, 23], and other staged attacks in power generators [16].

Therefore, the study of the effect and mitigation of attacks in CPSs has gained a great attention in recent years [21, 1, 6, 8, 25]. At the heart of CPSs is the network through which various components of the system exchange information. Hence, the analysis of attacks in the communication network, their detection, identification, and defense strategies are of major importance.

Recently, several researchers have studied the effect of attacks in the data communication of networked control systems [1, 10, 19, 32]. The work in [1, 10] focuses on the design of feedback controllers that minimize a control objective function. In this case, no delays are considered but only packet losses. The design of predictive controllers under delays and packet losses is proposed in [19], but out-of-sequence measurements are not explicitly considered in their solution. In [32], replay attacks are considered where the malicious node is able to replay old control messages that are sent to actuators.

In this paper, we devise a robust output-feedback controller which is resilient to an attack to the scheduling of packets in a networked

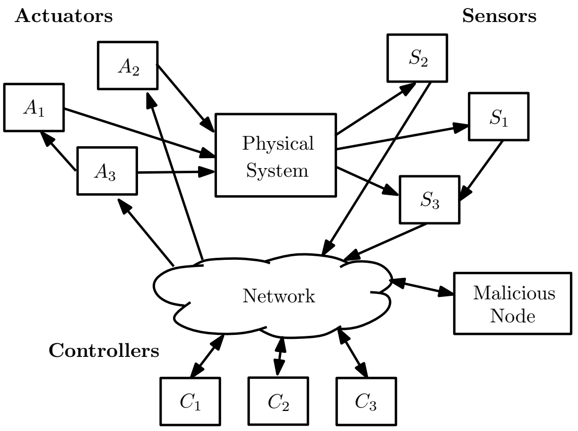

**Figure 1: Typical networked cyber-physical system with an adversarial attack on the shared network.**

control system. An attack on the scheduling algorithm will lead to time-varying delays and more importantly can lead to a change in the order by which packets are received. The proposed controller uses the available information (received packets up to the current time) so as to be robust with respect to such attack. Notwithstanding this fact, such controller has no worse performance than a controller that stores the received messages, reorders them, and presents them to the controller with a constant delay. Moreover, the proposed controller has better performance whenever there is no attack or the attacker does not launch the worse possible attack.

The rest of this paper is organized as follows. Section 2 introduces how a packet scheduling attack can be mounted and the assumptions on the attacker. Formal presentation of the problem along with the proposed controller and simulation results are presented in Sections 3 and 6 respectively. Finally, Section 7 concludes this paper.

## 2. PACKET SCHEDULING ATTACKS

In typical networked cyber-physical systems, multiple sensors send information to controllers through a shared communication channel, and controllers transmit control packets to actuators that are connected to the physical system. An illustration of a networked cyber-physical system is shown in Figure 1.

Data packets are scheduled for transmission and they must arrive to their destination node before a certain allowed deadline. In our work, we consider one controller that is co-located with the actuators. Accordingly, we will focus on attacks mounted only in the path between the multiple sensors and controller. We also assume that all the network nodes use cryptographic algorithms to encrypt, decrypt and authenticate packets. This prevents an adversary from changing the content of the packets. Replay attacks are also excluded in our scenario since packets are timestamped before encryption and the timestamps can be used to detect the replay of old data.

We consider stealth or covert attacks in which an attacker does not want the attack to be detected. One possible such attack consists of influencing the temporal characteristics of the network. It will result in time-varying delays and data packets possibly received out-of-order. However, to remain stealth, the attacker will not be able to delay the packets beyond a maximum allowable delay consistent with the network protocol in place.

This attack, if not addressed by the controller, can lead to unstable behavior. Figure 2 shows an example of a system with a classical Luenberger state observer and LQR controller (for formal definition of these terms please refer to [2]). In this example, the batch-reactor [27], a fourth order open-loop unstable system is simulated against both attacked and un-attacked packet schedules. The attack takes place in the path between from the sensor node to the controller node. Figure 3 shows how the packets are received where a cross sign indicates a message that is received out of its specified order and a circle indicates a packet received in the correct order. The height of the bars indicates the induced delay by the attacker. This example clearly shows that such attack on the packet scheduling can lead to instability.

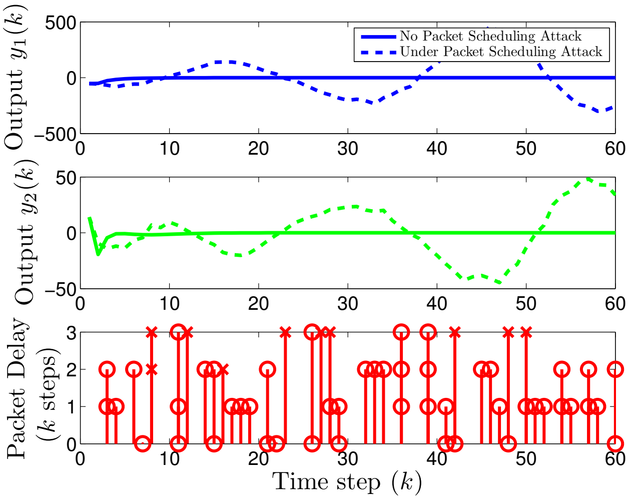

**Figure 2: Example showing that packet scheduling attack can render the closed loop system unstable. The upper figure shows the response of the system under the attack (red, dashed line), versus the nominal behavior in the absence of the attack (black, solid line). The bottom figure shows the effect of the adversary on the packet delivery, the cross mark denotes a packet delivered out of its order, while the circle denotes packet received in the correct order. Height of the bars denotes the delay induced in the each received packet.**

Packet scheduling attacks can be easily mounted to both wired and wireless communication channels using several techniques. A direct way of performing this attack is by an adversary placing malicious software on one of the packet routers in the path between sender and receiver.

Another way of mounting this attack is by means of resource unfairness attacks [7]. Unfairness is a weak type of Denial-of-Service (DoS) attacks. Both wired and wireless communication channels are exposed to unfairness attack. In wireless communication channel, an adversary can exhaust the shared communication channel by repeatedly sending packets leading to packet collision and automatic re-transmission, leading packets to miss their deadlines. Even wired networks are subject to unfairness attack. For example the arbitration mechanism of CAN bus can be easily attacked by adding a node to the bus which is able to flip just one bit in the identification part of the CAN packet leading to a maliciously dropping of specific packets from the network and firing automatic transmission by the CAN controller [17].

From this discussion, the effect of packet scheduling attacks can be seen as an attacker who can adversarially: 1) add a time-varying delay to the network and 2) alter the order by which packets are received by the controller. This scenario is illustrated in Figure 3.

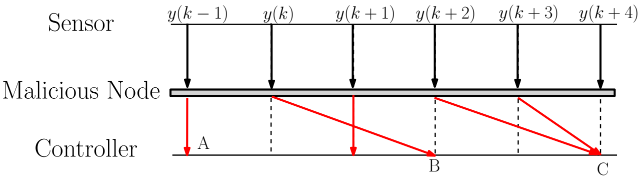

**Figure 3: An adversarial attack affecting the packet scheduling sent by sensors to the controller node can introduce time-varying delays leading to, A) packets delivered within the allowed deadline B)Packets delivered out of its order C)multiple packets received at the same time instance.**

The influence of the delay and missing data on the control system is a classic control analysis problem [9, 2]. Since the introduction of networked control systems, the analysis on the effect of fixed and time-varying delays as well as data loss on the control system has been the focus of much research [11, 12, 18, 31, 13, 29]. Even though the influence of short delays (lower than a sampling period) has been extensively studied, the effect of long delays and out-of-sequence messages has received less attention from the control community. The typical proposed solutions for a practical control system design under out-of-sequence messages is to: 1) utilize a buffer with length equal to the maximum expected message delay, thus avoiding any out-of-sequence issues [11, 12, 18] or 2) discard any out-of-sequence messages, assuming that the penalty for not using such packets is low [24, 13]. Such approaches may not be suitable since in case 1) a fixed delay is introduced in the system and no improvement is made when messages actually arrive with no delay, and in case 2) such strategy may discard a large amount of messages for persistently out-of-sequence messages.

This could be allowed for specific robust control system designs, but it is obvious that it may in general exhibit low control performances, depending on the delay values.

Another approach was proposed in [15] where optimal control under long delays and out-of-sequence measurements for linear stochastic systems is discussed. Even though out-of-sequence measurements are utilized to improve the state estimate, a new control actuation is not performed whenever this occurs. This assumption is not motivated by the authors and it can suffer the same drawbacks as the buffer approach. Furthermore, it is unclear what is the right statistics to be used in the case of an adversarial attack. A similar approach is described in [14]. Several researchers have looked at the problem of optimal estimation under out-of-sequence measurements [3, 30, 28] but with no consideration of the control system.

A direct over-designed controller can be implemented in this case by inserting a buffer at the controller node where all packets are stored, correctly reordered and then used by the controller after a fixed delay. We argue that an opportunistic design that takes care of the varying-delay and/or the out-of-sequence behavior can lead to better performance measured by means of a cost function. We formalize these assumptions in the next section.

## 3. MINIMAX CONTROL UNDER PACKET SCHEDULING ATTACK

In this paper, we consider a discrete-time linear time-invariant control system subject to both state and output disturbances. The dynamics of the system are described by:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + Dw_k, \\ y_k &= Cx_k + Ew_k, \end{aligned} \tag{3.1}$$

where $x_k \in \mathbb{R}^n$, $y_k \in \mathbf{Y} \subset \mathbb{R}^p$, $u_k \in \mathbf{U} \subset \mathbb{R}^m$, and $w_k \in \mathbf{W} \subset \mathbb{R}^d$ are the system state, output, control input, and disturbance input, all at time $k \in \mathbb{N}$ respectively. We have the following assumptions about the system and attacker capabilities:

(A1) There exists an upper bound $T$ for the packet delay in the underlying network.

(A2) A cryptographic protocol is used to encrypt, decrypt and authenticate the packets.

(A3) Packets are timestamped before being encrypted.

(A4) The attacker is capable of introducing time-varying delays on the packets. However, in order for the attacker to be kept stealthy, he will not delay a packet by more than $T$ time units.

The objective is to design a dynamic controller which is robust to the attack under consideration. It is beneficial to take the attacker's role to see how he should attack the system. The adversary by attacking the packet scheduling, i.e., by adding time-varying delays and/or altering the order of the packets, is actually preventing the controller from attenuating the effect of disturbances by preventing the controller from monitoring the exact state of the system. Hence, a natural defense strategy is to design a controller that is robust with respect to the worst disturbance input that is compatible with the received sequence of observations and the generated sequence of control actions. This leads to a minimax controller design using a dynamic game approach.

We follow the general framework of [4] to design a robust feedback controller using a zero-sum game-theoretic approach. The designed controller can be viewed as a dynamic game between two players. The controller is the minimizer player who tries to minimize the finite horizon quadratic cost (3.2) while the disturbance is the maximizer player.

$$\begin{aligned} J_\gamma(\mu,\nu) = {} & |x_{K+1}|^2_{Q_f} + \\ & \sum_{k=1}^{K} \left(|x_k|^2_Q + |u_k|^2 - \gamma^2 |w_k|^2\right) - \gamma^2 |x_1|^2_{Q_0}. \end{aligned} \tag{3.2}$$

Here, $\mu = \mu_1\mu_2 \dots \mu_k$ is the sequence of control inputs applied by the first player, $\nu = \nu_1\nu_2 \dots \nu_k$ is the sequence of disturbance inputs applied by the second player, $K$ is the finite horizon length, $x_1$ is the unknown initial state of the system, and $\gamma \in \mathbb{R}$ is the disturbance attenuation level. We use the notation $|.|$ to denote the Euclidean norm with positive definite weighting matrices $Q$, $Q_f$ and $Q_0$ of appropriate dimension.

The objective of the first player (controller) is to drive the state to zero while minimizing $J$. The objective of the second player (disturbance) and of the attacker is to increase the cost $J$ as much as possible. Note that when the game admits a solution, the controller ensures the following bound for the effect of the disturbance:

$$||\zeta|| \leq \gamma ||\sigma||, \tag{3.3}$$

where

$$||\zeta|| = |x_{K+1}|^2_{Q_f} + \sum_{k=1}^{K} \left(|x_k|^2_Q + |u_k|^2\right),$$

and

$$||\sigma|| = |x_1|^2_{Q_0} + \sum_{k=1}^{K} \left(|w_k|^2\right).$$

We show in this paper that this game obeys the conditions required for certainty equivalence [5] even under the varying-delay and/or the out-of-order messages imposed by the attacker. Under certainty equivalence, one can split the design problem into two parts: the first is to design an observer which estimates the worst possible state that matches the sequence of available inputs and outputs; the second is to design a controller which makes use of the estimated state in order to generate the new control input.

Following these steps, the worst case state estimator works as follows. Whenever the observer receives an out-of-order packet, it starts by reordering the set of previous $T$ messages, computes the worst case disturbance which is compatible with the available information, finds the corresponding worst case state estimate, and then plays the game as if the actual state was already located at this worst case estimate. The next two sections discuss the details of both the certainty equivalence property and how to construct the worst case observer and controller under the specified attack.

# 4. CERTAINTY EQUIVALENCE

Before we formulate the certainty equivalence property, we need to introduce some notation.

## 4.1 Notation

Recall that $u_k \in \mathbf{U}$ and $w_k \in \mathbf{W}$ denote the control input, and disturbance input at time $k$ respectively. We reserve the symbols $y$, $u$, and $w$ to denote the sequence of outputs, control inputs and disturbance inputs of finite length, i.e., $y \in \mathbf{Y}^*$, $u \in \mathbf{U}^*$, and $w \in \mathbf{W}^*$ respectively.

The overall disturbance $\omega$ is defined as the combination of the initial condition and the disturbance:

$$\omega := (x_1, w), \qquad \omega \in \Omega := \mathbb{R}^n \times \mathbf{W}^*. \tag{4.1}$$

We denote the solutions of (3.1) under the effect of sequences of inputs $u$ and disturbances $w$ as:

$$x_t = \phi_t(u, w, x_1), \tag{4.2}$$
$$y_t = \eta_t(u, w, x_1). \tag{4.3}$$

The conditions ensuring the certainty equivalence property are formulated in terms of the allowing information sets and its elements:

$$y^\tau \in \mathbf{Y}^\tau, \tag{4.4}$$
$$u^\tau \in \mathbf{U}^\tau, \tag{4.5}$$
$$w^\tau \in \mathbf{W}^\tau, \tag{4.6}$$
$$\omega^\tau \in \Omega^\tau = \mathbb{R}^n \times \mathbf{W}^\tau. \tag{4.7}$$

where $X^\epsilon$ denotes the $\epsilon$-fold cartesian product of $X$ with itself. Similarly we denote by $\eta^\tau$ to be the sequence of outputs $\eta_1 \eta_2 \ldots \eta_\tau$.

Let us consider the partial information problem. For any given integer $\tau \in \{1, 2, \ldots, k\}$ and sequence pair $(\bar{u}, \bar{y}) \in \mathbf{U}^\tau \times \mathbf{Y}^\tau$, we define the following subset $\Omega_\tau(\bar{u}, \bar{y})$ of $\Omega$:

$$\Omega_\tau(\bar{u}, \bar{y}) = \{\omega \in \Omega | \eta_k(\bar{u}, \omega) = \bar{y}_k, k = 1, \ldots, \tau\}, \tag{4.8}$$

which denotes all the disturbance sequences which are compatible with the input and output strings up to time $\tau$. We also introduce the following notation for the set of restrictions of $\Omega_\tau$:

$$\Omega_\tau^\tau(\bar{u}, \bar{y}) = \{\omega^\tau \in \Omega^\tau | \omega \in \Omega_\tau(\bar{u}, \bar{y})\}. \tag{4.9}$$

In the following discussion we drop the argument $(\bar{u}, \bar{y})$, however, one always needs to remember that $\Omega_\tau$ and its restriction are only the disturbance strings that are compatible with the observed sequences of inputs and outputs.

## 4.2 Certainty Equivalence

In this subsection, we review the the conditions under which certainty equivalence is known to hold.

### 4.2.1 Information Process

The controller (first player) does not have complete knowledge about the disturbance string nor about the system state. The observation process $\theta_\tau$ maps the observed inputs $u^\tau \in \mathbf{U}^\tau$ and outputs $y^\tau \in \mathbf{Y}^\tau$ to the set $\Omega_\tau$ of all disturbances compatible with these observations. Hence $\theta_\tau$ describes all the information about the disturbance that player 1 can extract from its observations. We note that $\theta_\tau$ satisfies the following properties:

- Consistency:

$$\forall u \in \mathbf{U}^\tau, \forall \omega \in \Omega, \forall \tau \in [1, K], \omega \in \Omega_\tau(u, \eta^\tau(u, \omega)). \tag{4.10}$$

- Perfect Recall:

$$\forall u \in \mathbf{U}^\tau, \forall \omega \in \Omega, \tau' > \tau \Rightarrow \Omega_{\tau'} \subset \Omega_\tau. \tag{4.11}$$

- Strict non-anticipativeness:

$$\forall u \in \mathbf{U}^\tau, \forall \omega \in \Omega, \forall \tau \in [1, K], \omega \in \Omega_\tau \Leftrightarrow \omega^{\tau-1} \in \Omega_\tau^{\tau-1}. \tag{4.12}$$

### 4.2.2 Assumption I:

The perfect-state information two-person zero-sum game where the disturbance has access to $u$, admits a state feedback saddle-point solution leading to the upper value function of the Isaacs equation:

$$\begin{aligned} V_k(x) &= \min_u \max_w V_{k+1}(Ax_k + Bu_k + Dw_k) + \\ &\quad + \left(|x_k|_Q^2 + |u_k|^2 - \gamma^2 |\omega_k|^2\right), \\ V_{K+1} &= |x_{K+1}|_{Q_f}^2. \end{aligned} \tag{4.13}$$

which represents the upper value of the game with performance index (3.2). Under this assumption, the minimum in $u$ is unique for every $(k, x)$.

### 4.2.3 Assumption II:

Introduce the following controller:

$$\hat{\mu}_\tau(\bar{u}^{\tau-1}, \bar{y}^{\tau-1}) := \mu_\tau^*(\hat{x}_\tau^\tau), \tag{4.14}$$

where, $\mu_\tau^*$ denotes the optimal controller strategy for the perfect state measurement problem and $\hat{\mu}_\tau$ is the controller which is based on the worst case estimate of the state $\hat{x}_\tau^\tau$ up to time $\tau$ based on the available string of inputs and observations $\bar{u}^{\tau-1}, \bar{y}^{\tau-1}$.

The control sequence generated by this controller is such that the following saddle point property holds for all $\omega \in \Omega$ and $\tau \in [1, K]$

$$\min_u \max_{\omega \in \Omega_{\tau-1}} G_\tau(\bar{u}^{\tau-1}.u, \omega) = \max_{\omega \in \Omega_{\tau-1}} \min_u G_\tau(\bar{u}^{\tau-1}.u, \omega), \tag{4.15}$$

where $G_\tau$ is the auxiliary performance index:

$$G_\tau(u^\tau, \omega^\tau) = V_{\tau+1}(x_{\tau+1}) + \sum_{k=1}^{K} \left(|x_k|_Q^2 + |u_k|^2 - \gamma^2 |\omega_k|^2\right). \tag{4.16}$$

### 4.2.4 Certainty Equivalence Principle:

For the partial information process and under assumptions I and II, the following problem has a solution for every $\tau$:

$$\max_{\omega \in \Omega_{\tau-1}} G_\tau(\bar{u}^{\tau-1}, \omega^{\tau-1}), \tag{4.17}$$

Moreover, the solution of this problem yields a uniquely defined minimax controller $\hat{\mu}_\tau(\bar{u}^{\tau-1}, \bar{y}^{\tau-1})$. This result means there exists a worst case disturbance that matches the string of inputs and partial observations available up to time $\tau$. Accordingly, the optimal minimax strategy for the first player is to construct a pair of controller and observer, where the controller is exactly the same as the optimal controller for the perfect state measurement, except it utilizes the state estimate instead of the measured state. The observer uses the information available up to time $\tau$ to estimate the worst case disturbance sequence and then use this information to estimate the worst case state trajectory which matches the string of inputs and outputs up to time $\tau$.

# 5. MINIMAX ESTIMATOR AND CONTROLLER DESIGN

We now switch the focus into how to utilize the certainty equivalence principle to design a worst case controller and observer which satisfy assumption II from the previous section. We will start by stating the following results proved in [4] upon which we base our results.

Consider a linear time invariant system subject to disturbance modeled with (3.1), and, along with the cost function (3.2). Suppose that at time $k \geq T$ only information up to time $k-T$ is available to the controller and $T$ is fixed. In other words, let's consider the information structure $\Omega_{\tau-T}$, this measurement process satisfies the three hypotheses (4.10)-(4.12). Accordingly, the auxiliary problem from Assumption II can be re-written as:

$$\max_{\omega^{\tau-1} \in \Omega^{\tau-1}_{\tau-T}} G_{\tau-1}(\bar{u}^{\tau-1}, \omega^{\tau-1}), \tag{5.1}$$

where

$$G_{\tau-1}(\bar{u}^{\tau-1}, \omega^{\tau-1}) = V_\tau(x_\tau) + \sum_{k=1}^{\tau-T} \left(|x_k|^2_Q + |u_k|^2 - \gamma^2|\omega_k|^2\right)$$
$$+ \sum_{\tau-T+1}^{\tau-1} \left(|x_k|^2_Q + |u_k|^2 - \gamma^2|\omega_k|^2\right),$$

which will result in an observer which is able to estimate the worst case disturbance (and thus the system state) which matches the strings of inputs and outputs up to time $\tau - T$. Then, by using forward dynamic programming, another observer can be used to estimate the worst case disturbance for the remaining time $[\tau - T + 1, \ldots \tau]$ where no observations are available to the controller.

Consider also the following dynamic controller which consists of the following controller/observer pair:

$$u_k = -B^T(M_{k+1}^{-1} + BB^T - \gamma^{-2}DD^T)^{-1} \cdot A(I - \gamma^{-2}\Sigma_k M_k)^{-1}\widetilde{x}_k \tag{5.2}$$

$$\widetilde{x}_{k+1} = A(I - \gamma^{-2}\widetilde{\Sigma}_k Q)^{-1}\widetilde{x}_k + Bu_k. \tag{5.3}$$

$$\hat{x}_{k+1} = A\hat{x}_k + Bu_k + A(\Sigma_k^{-1} + C^T N^{-1} C - \gamma^{-2}Q)^{-1} \cdot \left(\gamma^{-2}Q\hat{x}_k + C^T N^{-1}(y_k - C\hat{x}_k)\right). \tag{5.4}$$

where $N = E^T E$ and with initial conditions

$$\begin{cases} \hat{x}_1 = 0, \;\; \widetilde{x}_1 = 0, & \text{if } \tau \leq T \\ \widetilde{x}_{\tau-T+1} = \hat{x}_{\tau-T+1}, & \text{if } \tau > T \end{cases} \tag{5.5}$$

where $\hat{x}_k$ represents the closed-loop observer which incorporates the messages being received to update the current state estimate, while $\widetilde{x}_k$ represents the open-loop observer which runs over the period with no messages being received while taking into account the worst case disturbance.

Additionally, $M$ and $\Sigma$ are the solutions of the following Game Algebraic Riccati Equation (GARE):

$$M_k = A^T(M_{k+1}^{-1} + BB^T - \gamma^{-2}DD^T)^{-1}A + Q^TQ, \quad M_f = Q_f \tag{5.6}$$

$$\Sigma_{k+1} = A(\Sigma_k^{-1} + C^T N^{-1} C - \gamma^{-2}Q)^{-1}A^T + DD^T, \quad \Sigma_1 = Q^{-1} \tag{5.7}$$

$$\widetilde{\Sigma}_{k+1} = A(\widetilde{\Sigma}_k^{-1} - \gamma^{-2}Q)^{-1}A^T + DD^T, \tag{5.8}$$

where $\widetilde{\Sigma}_{k+1}$ has initial conditions

$$\begin{cases} \widetilde{\Sigma}_1 = Q^{-1} & \text{if } \tau \leq T \\ \widetilde{\Sigma}_{\tau-T+1} = \Sigma_{\tau-T+1} & \text{if } \tau > T \end{cases} \tag{5.9}$$

PROPOSITION 5.1 (THEOREM 6.6 IN [4]). *The controller and observer pair described by* (5.2)-(5.9)*, guarantees that minimax control for* (3.1)*, with quadratic cost given by* (3.2)*, achieves achieves the performance level of* $\gamma$ *if the following conditions are satisfied:*

- *The Riccati equations* (5.6) *and* (5.8) *must admit a solution over* $[1, K+1]$*, and the Riccati equation* (5.7) *has a solution over* $[1, K-T+1]$.
- *The solution of the Observer Riccati equation* (5.8) *must satisfy:*

$$\rho(\widetilde{\Sigma}_k Q) < \gamma^2, \;\; k = 1, ..., K+1. \tag{5.10}$$

- *The solution of the two Riccati* (5.6) *and* (5.8) *equations must satisfy :*

$$\rho(\widetilde{\Sigma}_{k+1} M_{k+1}) < \gamma^2, \;\; k = 1, ..., K. \tag{5.11}$$

This proposition show that this observer is able to always estimate the worst case state by using the information received up to time $k - T$ and then run a worst case open-loop observer for the period where no information is available.

Now we can generalize this result into the case of networked system under the specified attack.

## 5.1 Minimax Estimator and Controller under Packet Scheduling Attack

In order to design a minimax controller, we need first to check that all the assumptions required for certainty equivalence hold. First, lets examine the information structure presented in this situation. The packet scheduling attack affects mainly the information process and the amount of information presented at the controller at each time instance.

We can define the following set:

$$\mathbf{T} = \{k \in \{1, \ldots, \tau\} | \text{packet is received and has timestamp} = k\} \tag{5.12}$$

The information structure for the system under attack can be then defined as:

$$\widetilde{\Omega}_\tau(\bar{u}, \bar{y}) = \{\omega \in \Omega | \eta_k(\bar{u}, \omega) = \bar{y}_k, k \in \mathbf{T}\}, \tag{5.13}$$

since packets are timestamped and can be reordered according to the time-stamps, it is not difficult to see that this information structure satisfies assumptions (4.10)-(4.12).

For the rest of the certainty equivalence assumptions, assumption I is not related to the packet reception behavior and thus it follows directly if the system without the attack does satisfy this assumption, thus the system under the attack satisfies them as well. Assumption II requires that the designed controller and observer satisfies the saddle point equation shown. This will be discussed in the remaining of this section.

Consider now the auxiliary problem defined in Assumption II. For the system under attack, the auxiliary problem can be written as:

$$\max_{\omega^{\tau-1}\in\tilde{\Omega}^{\tau-1}_{\tau-1}} G_{\tau-1}(\bar{u}^{\tau-1},\omega^{\tau-1}), \tag{5.14}$$

where

$$G_{\tau-1}(\bar{u}^{\tau-1},\omega^{\tau-1}) = V_\tau(x_\tau) + \sum_{k\in\mathbf{T}}\left(|x_k|^2_Q + |u_k|^2 - \gamma^2|\omega_k|^2\right) + \sum_{k\notin\mathbf{T}}\left(|x_k|^2_Q + |u_k|^2 - \gamma^2|\omega_k|^2\right).$$

With again the same intuition of having an observer which estimates the worst case disturbance over the time slots where the controller has information and then runs an open-loop (forward dynamic programming) to estimate the worst case for the remaining time where no information is available. Algorithm 1 utilizes this intuition.

As define above, $\Sigma_k$ is used when there is a measurement reception, and $\widetilde{\Sigma}_k$ whenever the system is in open-loop and no measurement is received at the controller. We define $\bar{\Sigma}_k$ as taking the value of either $\Sigma_k$ or $\widetilde{\Sigma}_k$ in the case of reception or no reception, respectively, of a message. An auxiliary state is also introduced as $\bar{x}_k$ which follows the same definition depending on the reception or no reception of a measurement. Additionally, we define the time step at which a packet is transmitted as $t_k$. The current time is denote as $\kappa$. The number of packets received at each time interval is denoted as $N_{pkts}$. Since multiple packets may arrive during the last time interval, we denote $t^i_k$ as the time of each packet $i \in [1, N_{pkts}]$. We also define the buffers, $\Theta_x \in \mathbb{R}^{(n,T+1)}$, $\Theta_u \in \mathbb{R}^{(m,T+1)}$, $\Theta_y \in \mathbb{R}^{(p,T+1)}$ and $\Theta_\Sigma \in \mathbb{R}^{(n,n(T+1))}$, which store the values of $\widetilde{x}_k$, $u_k$, $y_k$ and $\bar{\Sigma}_k$ in the interval $[\kappa - T - 1, \kappa]$. Values are stored in ascending order of transmission time and if a measurement does not arrive at the controller at a particular time $k$, then $\Theta_y(k) \in \emptyset$ (empty). We summarize this discussion in the following result:

THEOREM 5.2. *The implementation of the dynamic observer and controller pair shown in algorithm 1 guarantees that the minimax controller achieves the performance level of $\gamma$ for the system under the packet scheduling attack, described by the assumptions (A1)-(A4), if the same conditions of proposition 5.1 are satisfied.*

PROOF. Algorithm 1 utilizes the same controller as proposition 5.1. Thus, with the conditions of Certainty Equivalence being satisfied, it is sufficient to show that the observer described in algorithm 1 constructs the worst case state estimate under the information pattern imposed by the packet scheduling attack.

Define $y^\tau(\Delta)$ to be:

$$y^\tau(\Delta) = \begin{cases} y_i, & i = 1,\ldots\tau-\Delta \\ 0, & i > \tau - \Delta \end{cases} \tag{5.15}$$

proposition 5.1 guarantees that the observer defined in (5.3) and (5.4) estimates the worst case state for the described output string $y^\tau(T)$. For the system under the attack, at each time $k$ the observer reorders the packets in its correct order. The string of outputs can be shown as a concatenation of multiple $y^\tau(\Delta)$:

$$y_k = y^{\tau_1}(\Delta_1)y^{\tau_2}(\Delta_2)\ldots y^{\tau_n}(\Delta_n). \tag{5.16}$$

**Algorithm 1** Minimax control under packet scheduling attack

```
Define $N_{pkts}$ and $\Theta_y$ based on the packets received between $[\kappa-1,\kappa]$.
if $N_{pkts} = 0$ then                                      ▷ No packet received
    $\bar{x}_{k+1} \leftarrow$ (5.3)
    $\bar{\Sigma}_{k+1} \leftarrow$ (5.8)
else                                                        ▷ Packet(s) have received
    $t_k^{min} \leftarrow \min_{i=1,\ldots,N_{pkts}}(t^i_k)$
    $\bar{x}_{k+1} \leftarrow$ (5.5),                        ▷ Initializations
    with initial $\bar{x}_{t_k^{min}-1} \leftarrow \Theta_x(\kappa - t_k^{min} - 1)$
    $\bar{\Sigma}_{k+1} \leftarrow$ (5.9)
    with initial $\bar{\Sigma}_{t_k^{min}-1} \leftarrow \Theta_\Sigma(\kappa - t_k^{min} - 1)$
    for $k = t_k^{min} - 1 : \kappa$ do                     ▷ Re-compute $\bar{x}$ and $\bar{\Sigma}$
        $u_k \leftarrow \Theta_u(\kappa - k)$
        $y_k \leftarrow \Theta_y(\kappa - k)$
        if $\Theta_y(k) \in \emptyset$ then                  ▷ No packet arrival
            $\bar{x}_{k+1} \leftarrow$ (5.3)
            $\bar{\Sigma}_{k+1} \leftarrow$ (5.8)
        else                                                ▷ If packet arrived
            $\bar{x}_{k+1} \leftarrow$ (5.4)
            $\bar{\Sigma}_{k+1} \leftarrow$ (5.7)
        end if
    end for
end if
$M_k \leftarrow$ (5.8)
$u_k \leftarrow$ (5.2)
```

The observer constructed in algorithm 1 can be shown as multiple runs of the observer shown in proposition 5.1 over each $y^{\tau_j}(\Delta_j)$ separately, where $\Sigma_k$ and $\widetilde{\Sigma}_k$ are now defined as $\bar{\Sigma}_k$. □

## 5.2 Memory and computation requirements

The proposed estimator and controller under packet scheduling attack requires the storage of the information $\Theta_x$, $\Theta_u$, $\Theta_y$ and $\Theta_\Sigma$, at the controller/estimator unit as detailed in algorithm 1. In the case of the usage of a buffer approach, as discussed in Section 2, the implementation requires the storage of the same information as the proposed solution. This is the case as this approach computes equations (5.2)-(5.8) which require the same information set. Additionally, both implementations require the storage of the solution of the Riccati equation $M_k$.

With respect to the computational complexity, while the proposed controller/estimator requires de online computation of $\bar{\Sigma}$, in the buffer case this value can be pre-computed and stored in memory. However, this would increase the memory requirements when compared to the propose solution.

# 6. SIMULATION RESULTS

We now illustrate our results using a numerical example to validate the controller and estimator proposed in Section 5. The proposed solution is used for the control of the Batch Reactor process from [27] which is a fourth order unstable linear system with two inputs with system parameters defined as

$$A = \begin{pmatrix} 1.38 & -0.2077 & 6.715 & -5.676 \\ -0.5814 & -4.29 & 0 & 0.675 \\ 1.067 & 4.273 & -6.654 & 5.893 \\ 0.048 & 4.273 & 1.343 & -2.104 \end{pmatrix}$$

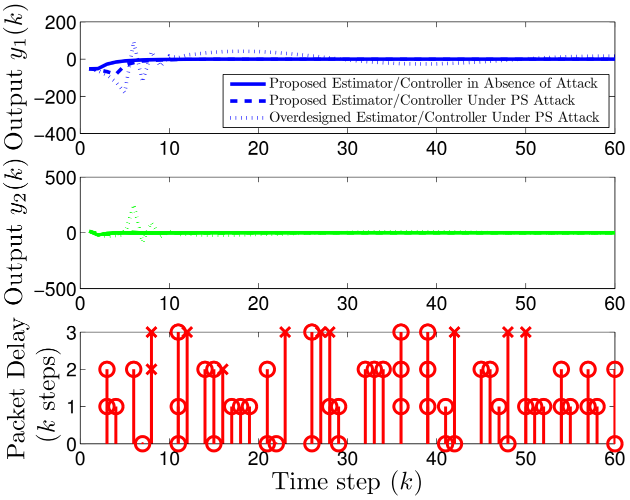

**Figure 4: The upper figure shows the output value of the system under the three different cases. Lower figure depicts the packet delay induced by the attacker to each *received* packet.**

$$B = \begin{pmatrix} 0 & 0 \\ 5.679 & 0 \\ 1.136 & -3.146 \\ 1.136 & 0 \end{pmatrix}, \ C \begin{pmatrix} 1 & 0 & 1 & -1 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \ D = \mathbf{0},$$

The worst-case process disturbance $w(k)$ for this system is defined in [4] and we use it also in our example. Additionally, the maximum allowed delay in the network is set to $T = 4$. The attacker will pick any delay between $[0, T]$ to affect the transmitted packet. Moreover, we select $Q_f$ and $Q$ as the identity matrix.

We evaluate the proposed solution in the absence of an attack and under a packet switching attack, and we compare it to the case where an over-designed estimator/controller would be used. The over-designed estimator/controller is the typical buffer case discussed in Section 2. The buffer length is defined to be the maximum delay $T = 4$.

Both the over-designed and our proposed estimator/controller are designed using a minimax approach where $\gamma$ is picked to solve the algebraic Riccati equations (5.7)-(5.8) for the maximum delay. We set $\gamma = 36$ which is the minimum value which renders the system stable under the buffer implementation.

Figure 4 shows the time-response analysis for a 60 step simulation of the closed-loop control system, respectively. The upper figure shows the output value of the system under the three different cases, and lower figure depicts the packet delay induced by the attacker to each *received* packet. As the proposed controller and state-estimator makes use of all information up to time the current time step, an improved performance with respect to the over-designed implementation is obtained as shown in Figure 4.

The final cost in (3.2) obtained for the proposed minimax approach is 2.03% higher than the un-attacked scenario while the over-designed controller pays 103.3% more cost than the un-attacked scenario. This result reflects the opportunistic nature of our design. Figure 5 depicts the evolution of the logarithm of the control cost $\log J(k)$ under all cases.

## 7. CONCLUSIONS AND FUTURE WORK

In this paper, we have introduced a minimax defense for packet scheduling attacks. The main technical point is to show that the certainty equivalence property holds under the varying-delay and out-of-order information structure. It is then straight forward to design a worst case state-estimator and controller under this information structure. The final design is of opportunistic nature in the sense that it is designed for the worst case delay while immediately using the information in the received packets. The ability to immediately use the received information leads to better performance as measured by a quadratic cost. Simulation results show the feasibility of the proposed design.

As future work, we could generalize the results to the case where the attacker can mount attack on the path between the controller and the actuator. Another research direction is investigating the case where a system has more than one actuator, each actuator is controlled by a separate controller and the packet scheduling attack forces packets to arrive at different controllers in a different order. Such attack will lead to controllers with different knowledge about the system, yet they still need to be resilient.

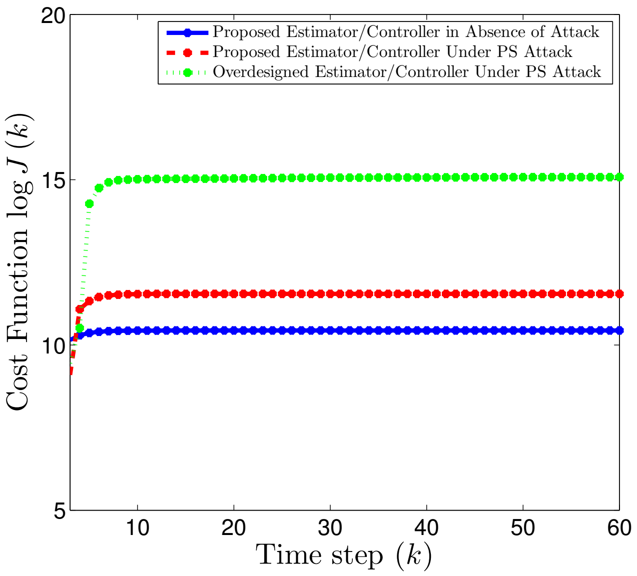

**Figure 5: Evolution of the control cost $\log J(k)$ for each of the evaluated cases.**

## 8. ADDITIONAL AUTHORS

## 9. REFERENCES

[1] S. Amin, A. A. Cardenas, and S. Sastry. Safe and secure networked control systems under denial-of-service attacks. In R. Majumdar and P. Tabuada, editors, *Hybrid Systems: Computation and Control*, volume 5469 of *Lecture Notes in Computer Science*, pages 31–45. Springer Berlin Heidelberg, 2009.

[2] K. Åström and B. Wittenmark. *Computer controlled systems*. Prentice Hall Englewood Cliffs, NJ, 1990.

[3] Y. Bar-Shalom. Update with out-of-sequence measurements in tracking: exact solution. *Aerospace and Electronic Systems, IEEE Transactions on*, 38(3):769 – 777, jul 2002.

[4] T. Başar and P. Bernhard. *H-Infinity Optimal Control and Related Minimax Design Problems: A Dynamic Game Approach*. Modern Birkhäuser Classics. Birkhäuser Boston, 2008.

[5] P. Bernhard. A discrete-time min-max certainty equivalence principle. *Systems and Control Letters*, 24(4):229 – 234, 1995.

[6] A. A. Cárdenas, S. Amin, Z.-S. Lin, Y.-L. Huang, C.-Y. Huang, and S. Sastry. Attacks against process control systems: risk assessment, detection, and response. In *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*, ASIACCS '11, pages 355–366, New York, NY, USA, 2011. ACM.

[7] X. Chen, K. Makki, K. Yen, and N. Pissinou. Sensor network security: a survey. *Communications Surveys Tutorials, IEEE*, 11(2):52 –73, quarter 2009.

[8] H. Fawzi, P. Tabuada, and S. Diggavi. Secure state-estimation for dynamical systems under active adversaries. In *Communication, Control, and Computing (Allerton), 2011 49th Annual Allerton Conference on*, pages 337 –344, sept. 2011.

[9] T. Glad and L. Ljung. *Control Theory: Multivariable and Nonlinear Methods*. Taylor & Francis, 2000.

[10] A. Gupta, C. Langbort, and T. Başar. Optimal control in the presence of an intelligent jammer with limited actions. In *Decision and Control (CDC), 2010 49th IEEE Conference on*, pages 1096 –1101, dec. 2010.

[11] Y. Halevi and A. Ray. Integrated communication and control systems. I - analysis. *ASME, Transactions, Journal of Dynamic Systems, Measurement and Control.*, 110:367 – 373, December 1988.

[12] Y. Halevi and A. Ray. Integrated communication and control systems. II - design considerations. *ASME, Transactions, Journal of Dynamic Systems, Measurement and Control.*, 110:367 – 373, December 1988.

[13] J. P. Hespanha, P. Naghshtabrizi, and Y. Xu. A survey of recent results in networked control systems. *Proceedings of the IEEE*, 95(1):138–162, 2007.

[14] H. Hirano, M. Mukai, T. Azuma, and M. Fujita. Optimal control of discrete-time linear systems with network-induced varying delay. In *American Control Conference, 2005. Proceedings of the 2005*, pages 1419 – 1424 vol. 2, june 2005.

[15] B. Lincoln and B. Bernhardsson. Optimal control over networks with long random delays. In *Proceedings of the International Symposium on Mathematical Theory of Networks and Systems*, Jan. 2000.

[16] J. Meserve. Staged cyber attack reveals vulnerability in power grid, Sept. 2007.

[17] M. Mostafa, M. Shalan, and S. Hammad. FPGA-based low-level can protocol testing. In *System-on-Chip for Real-Time Applications, The 6th International Workshop on*, pages 185–188. IEEE, 2006.

[18] J. Nilsson. *Real-Time control systems with delays*. PhD thesis, Lund Institute of Technology, Jan 1998. Ph.D. thesis.

[19] Z. H. Pang, G. P. Liu, and Z. Dong. Secure networked control systems under denial of service attacks. In *18th IFAC World Congress*, 2011.

[20] R. Poovendran, K. Sampigethaya, S. K. S. Gupta, I. Lee, K. V. Prasad, D. Corman, and J. Paunicka. Special issue on cyber - physical systems [scanning the issue]. *Proceedings of the IEEE*, 100(1):6 –12, jan. 2012.

[21] S. Radosavac, A. A. Cárdenas, J. S. Baras, and G. V. Moustakides. Detecting ieee 802.11 mac layer misbehavior in ad hoc networks: Robust strategies against individual and colluding attackers. *J. Comput. Secur.*, 15(1):103–128, Jan. 2007.

[22] M. Raya and J. Hubaux. Securing vehicular ad hoc networks. *Journal of Computer Security*, 15(1):39–68, 2007.

[23] T. Rid. Cyber war will not take place. *Journal of Strategic Studies*, 2011.

[24] P. L. Tang and C. de Silva. Compensation for transmission delays in an ethernet-based control network using variable-horizon predictive control. *Control Systems Technology, IEEE Transactions on*, 14(4):707 – 718, july 2006.

[25] A. Teixeira, D. Pérez, H. Sandberg, and K. H. Johansson. Attack models and scenarios for networked control systems. In *Proceedings of the 1st international conference on High Confidence Networked Systems*, HiCoNS '12, pages 55–64, New York, NY, USA, 2012. ACM.

[26] J. Vijayan. Stuxnet renews power grid security concerns, Jul. 26 2010.

[27] G. Walsh, H. Ye, and L. Bushnell. Stability analysis of networked control systems. *IEEE Transactions on Control Systems Technology*, pages 2876–2880, 1999.

[28] A. Westenberger, B. Duraisamy, M. Munz, M. Muntzinger, M. Fritzsche, and K. Dietmayer. Impact of out-of-sequence measurements on the joint integrated probabilistic data association filter for vehicle safety systems. In *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pages 438 –443, june 2012.

[29] T. Yang. Networked control system: a brief survey. *Control Theory and Applications, IEE Proceedings -*, 153(4):403 – 412, july 2006.

[30] K. Zhang, X. Li, and Y. Zhu. Optimal update with out-of-sequence measurements. *Signal Processing, IEEE Transactions on*, 53(6):1992 – 2004, june 2005.

[31] W. Zhang, M. S. Branicky, and S. M. Phillips. Stability of networked control systems. *IEEE Control Systems Magazine*, 21(1):84 –99, Feb. 2001.

[32] M. Zhu and S. Martinez. On resilient consensus against replay attacks in operator-vehicle networks. In *American Control Conference (ACC), 2012*, pages 3553 –3558, june 2012.