52nd IEEE Conference on Decision and Control
December 10-13, 2013. Florence, Italy

A model predictive approach for cyber-attack detection and mitigation in control systems

Albert Rosich, Holger Voos, Yumei Li and Mohamed Darouach

***Abstract*— The paper presents a new approach for control security. Specifically, cyber-attacks on the controller are investigated by means of optimization techniques in order to determine the worst-case scenario. Then, a novel attack detector based on limit checking is introduced. The particularity of this detector is that no specific controller knowledge is necessary. Hence, the vulnerability of the detector can be reduced since no reconfiguration is required (limited accessibility). Finally, the paper shows that the effect of the attacks on the system can be significantly mitigated by applying proper optimal control laws.**

I. Introduction

Nowadays, control devices are interconnected trough the process control network which, in turn, is connected to external networks such as Internet. This poses several problems in terms of security, especially when the process under control is safety critical. In fact, several cases of cyber-attacks on critical controlled plants have already been reported (e.g. [1], [2]). As pointed out in [3], the traditional tools from computer security may not be sufficient or suitable for controlled systems. Therefore, novel methods exploiting knowledge of the controller and the plant need to be developed [4], [5]. Recent works address the security on control systems by means of control-based techniques. For instance, new control schemes are designed in [6] and [7] in order to handle different attacks.

Up to now, the attack scenarios considered in control system security are based on plant-controller communication disruption. In a networked control system, the sent and received data in the control loop is blocked or modified by the attacker. See [8] for a good overview on attack scenarios. In this paper, a new attack scenario is studied. Based on the Stuxnet case [1] where a controller was reprogrammed, we specially focus on the case wherein the controller is hijacked an a harmful control signal is injected into the control loop.

A typical approach to deal with cyber-attacks, adopted by the majority of the works devoted to control system security, is to implement attack detectors based on system models (e.g. [9], [10]). Inspired by fault diagnosis detectors [11], [12], the basic idea is to use a model of the system in order to validate whether the actual system behavior corresponds to the attack-free behavior, or else to some sort of attack. Usually, this is applied to a dynamic process where model uncertainties are handled by observer design techniques [13]. It should not be difficult, since a controller is indeed a dynamic system, to apply the same approach for cyber-attacks targeting the controller. A representative block diagram of this approach applied to an attacked controller is shown in Fig. 1. It should be noted that this approach must provide good results since reliable, if not exact, controller models are easy to obtain. However, this poses some security problems as it will be next discussed.

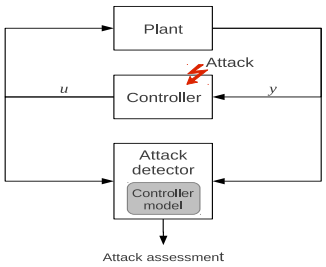

Fig. 1. Basic approach to detect attacks on the controller.

One of the largest advantages of having controllers connected to the network is that they can be remotely reconfigured or tuned by control and plant engineers. By adopting the mentioned approach (depicted in Fig. 1), the detector should be also modified accordingly every time the controller is modified. Consequently, the same level of accessibility needs to be available for both the controller and the detector, which in turn implies that the controller vulnerabilities are somehow inhered by the detector. Therefore, there are good reasons to believe that this approach is not suitable since potential attacks could target both, the controller and the detector, at the same time.

To face this problem, this paper presents a novel approach for detecting attacks on the controller with the particularity that no prior controller information is needed (no controller model-based detector). Therefore, this new detector does not need to be adjusted every time that the controller is modified. Consequently, the detector can be implemented with a very limited accessibility (e.g. no outside access or reconfiguration are allowed), and thus with a high protection degree against attacks. For these reasons, it will be assumed that the detector can not be attacked in this case.

In this paper, safety limits on the measured variables will be considered to validate whether the plant is working

This work was supported by the Fonds National de la Recherche, Luxembourg, under the project CO11/IS/1206050 (SeSaNet)

A. Rosich, H. Voos and Y. Li are with the Interdisciplinary Centre for Security Reliability and Trust (SnT), University of Luxembourg, Luxembourg. albert.rosich@uni.lu, holger.voos@uni.lu, yumei.li@uni.lu.

M. Darouach is with the Centre de la Recherche en Automatique de Nancy (CRAN), Université de Lorraine, France. mohamed.darouach@univ-lorraine.fr.

properly. The limit checking technique is widely used in industry to detect malfunctions and problems, since it is simple and free of models [14]. The limits are typically fixed beforehand, and then an alarm is triggered when one of the measurement crosses its corresponding limit. In addition, it will be assumed that a backup controller is available, thus the hijacked controller can instantaneously be replaced by the backup controller.

Besides the new attack detector scheme, the paper also presents a preliminary study of the worst-case attack as well as an optimal control with the aim to mitigate the attack effect once it has been detected. This study is based on the system model and carried out by means of optimization techniques. It is important to emphasize that no model uncertainty, and neither perturbation, is considered. Consequently, no feedback is taken into consideration. In case of necessity, it should not be difficult to overcome this lack of feedback by extending the present study to constrained optimal control techniques such as model-predictive control [15] or multiparametric optimal control [16]. The objective of this study is twofold: 1) demonstrate that the ideal case of a backup controller is not sufficient to handle attacks, and 2) motivate that significant improvements can be obtained if suitable optimal controls are used.

II. Illustrative Example

A small and simple example will be used throughout the paper with the objective to have a better understanding of the context. The results provided by this example are quite intuitive and highlight the benefits of the proposed approach. Nevertheless, the approach is not limited to this example and can be straightforwardly extended to more complex and larger systems.

The example consists of a one-dimensional moving cart whose horizontal position needs to be controlled (see Fig. 2). The system obeys the Newton's second law and is modeled in continuous time as

$$\frac{d^2y(t)}{dt^2} = u(t)$$

where unitary mass is choosen for simplicity. The corresponding discrete state space model for this system is

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 1 \cdot 10^{-2} \\ 5 \cdot 10^{-3} \end{bmatrix} u(k) \tag{1a}$$

$$y(k) = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} \tag{1b}$$

where x_1 and x_2 represent the velocity and the position of the cart, respectively. The sample time of both, the cart system and the controller is set to $T_s = 0.1$ seconds. It is important to point out that the actuator range is limited to the interval $[u^- \quad u^+] = [-1 \quad 1]$ and the cart should move inside the position interval $[y^- \quad y^+] = [-5 \quad 5]$ for safety reasons.

A PID controller is used to control the cart position whereas a standard monitoring system checks that the position does not exceed the limits. The PID controller is

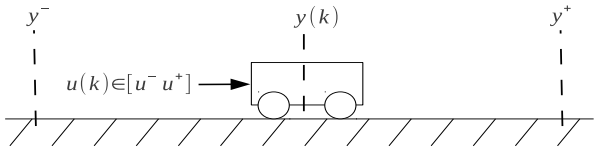

Fig. 2. Cart system.

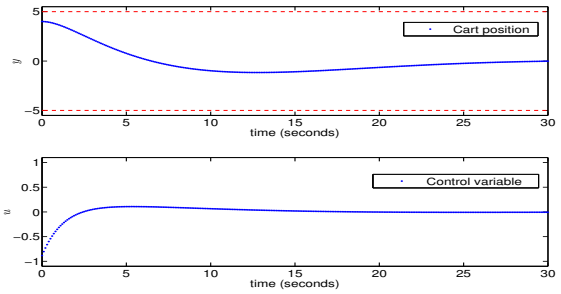

Fig. 3. Typical cart and PID controller responses.

represented by the following discrete transfer function

$$u(k) = \left(K_P + K_I \frac{T_s}{z-1} + K_D \frac{z-1}{z-(1-Ts)}\right) e(k) \tag{2}$$

where $K_P = 2.48 \cdot 10^{-2}$, $K_I = 9.35 \cdot 10^{-5}$ and $K_D = 1.96 \cdot 10^{-1}$ are the proportional, integral and derivative coefficients. The error input signal is fed back as $e(k) = -y(k)$, since the set-point position is 0.

Figure 3 shows typical system responses when cart initially rests at $y(0) = 4$ and the PID controller drives it to the zero position. Note that no limit is crossed.

III. Attack Analysis

Before presenting the attack detector, the worst-case attack scenario is analyzed in this section by means of optimization techniques.

Assume that an attacker hijacks the controller and is able to replace the control variable by any other signal of length N_a. Without a loss of generality, it will be also assumed that the attack starts at time instant $t = 0$, i.e. $u_a(0), \ldots, u_a(N_a - 1)$, and it will be detected at $t = N_a$. The goal of the attacker is to compromise system security by forcing a measured variable to cross the safety limit and to violate the limits as far as possible while satifying the actuator ranges at the same time. Furthermore, for the sake of simplicity, it is assumed that once a limit is crossed, the attack is instantaneously rejected and a backup controller with same structure and parameters as the attacked controller is connected to the system in order to recover it. Hence, the attack analysis can be divided into two parts:

1) *Attack perpetration*: The attacker performs the attack by introducing a signal u_a while it remains undetected. The system response for this part can be calculated by

$$x(k+1) = Ax(k) + Bu_a(k) \tag{3a}$$
$$y(k) = Cx(k) + Du_a(k) \tag{3b}$$

for $k = \{0, \cdots, N_a - 1\}$.

2) *Attack effect*: The attack has already been detected and rejected which means that now the system is driven by the back up controller. In this case, a state feedback control law $u(k) = -Kx(k)$ is chosen to represent a generic backup controller. A time horizon of N_e steps has been considered for the system response in this part. Thus, the system can be represented by the following autonomous system,

$$x(k+1) = (A - BK)x(k) \tag{4a}$$
$$y(k) = Cx(k) \tag{4b}$$

for $k = \{N_a + 1, \cdots, N_a + N_e - 1\}$

Taking into account (3) and (4), the worst attack scenario is computed by means of the following optimization problem,

$$\max_{u_a(0),\ldots,u_a(N_a-1)} y_i(N_a + N_e - 1)^T y_i(N_a + N_e - 1) \tag{5a}$$

subject to:

$$x(k+1) = Ax(k) + Bu_a(k) \tag{5b}$$
$$y(k) = Cx(k) + Du_a(k) \tag{5c}$$
$$\text{for } k = \{0, \cdots, N_a - 1\}$$
$$x(k+1) = (A - BK)x(k) \tag{5d}$$
$$y(k) = Cx(k) \tag{5e}$$
$$\text{for } k = \{N_a, \cdots, N_a + N_e - 1\}$$
$$u_a(0), \ldots, u_a(N_a - 1) \in [u^- \quad u^+] \tag{5f}$$
$$y(0), \ldots, y(N_a - 1) \in [y^- \quad y^+] \tag{5g}$$

The solution of this optimization problem ensures a maximum value of the measured variable $y_i \in y$ at time instant $t = N_a + N_e - 1$.

The constraints of the optimization problem in (5) can straightforwardly be rewritten by extending (3a) and (4a). Hence, it holds that

$$x(k) = A^k x(0) + \begin{bmatrix} A^{k-1}B & \cdots & AB & B \end{bmatrix} \begin{bmatrix} u_a(0) \\ \vdots \\ u_a(k-1) \end{bmatrix} \tag{6}$$

for any $k \in \{0, \cdots, N_a - 1\}$, and

$$x(k) = (A - BK)^{k - N_a} x(N_a) \tag{7}$$

for any $k \in \{N_a, \cdots, N_a + N_e - 1\}$. From (4b), (6) and (7), the outputs of the system at time $t = N_a + N_e - 1$ can be computed as

$$y(N_a + N_e - 1) = \mathcal{P}x(0) + \mathcal{Q}\mathcal{U}_a \tag{8}$$

where

$$\mathcal{P} = C\big(A - BK\big)^{N_e - 1} A^{N_a}$$
$$\mathcal{Q} = C\big(A - BK\big)^{N_e - 1} \begin{bmatrix} A^{N_a - 1}B & \cdots & AB & B \end{bmatrix}$$
$$\mathcal{U}_a = \begin{bmatrix} u_a(0) & \cdots & u_a(N_a - 1) \end{bmatrix}^T$$

Similarly, the system outputs for $k = \{0, \cdots, N_a - 1\}$, i.e. $\mathcal{Y} = [y(0) \ \cdots \ y(N_a - 1)]^T$, can be computed as

$$\mathcal{Y} = \mathcal{G}_x x(k) + \mathcal{G}_u \mathcal{U}_a \tag{9}$$

where

$$\mathcal{G}_x = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{N_a - 1} \end{bmatrix}, \ \mathcal{G}_u = \begin{bmatrix} D & 0 & \cdots & 0 \\ CB & D & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ CA^{N_a - 2}B & CA^{N_a - 3}B & \cdots & D \end{bmatrix}$$

Finally, the optimization problem in (5) can be reformulated as a standard quadratic optimization problem, i.e.

$$\max_{\mathcal{U}_a} \quad \mathcal{U}_a^T \mathcal{Q}_i^T \mathcal{Q}_i \mathcal{U}_a + 2x(0)^T \mathcal{P}_i^T \mathcal{Q}_i \mathcal{U}_a + x(0)^T \mathcal{P}_i^T \mathcal{P}_i x(0) \tag{10a}$$

subject to:

$$\begin{bmatrix} -\mathcal{G}_u \\ \mathcal{G}_u \\ -I \\ I \end{bmatrix} \mathcal{U}_a \leq \begin{bmatrix} -\mathcal{Y}^- + \mathcal{G}_x x(k) \\ \mathcal{Y}^+ - \mathcal{G}_x x(k) \\ -\mathcal{U}^- \\ \mathcal{U}^+ \end{bmatrix} \tag{10b}$$

where $\mathcal{P}_i$ and $\mathcal{Q}_i$ denote the ith-row of the matrices $\mathcal{P}$ and $\mathcal{Q}$. Matrix I is an identity matrix of the dimension $(N_a - 1)$ while $\mathcal{Y}^-$, $\mathcal{Y}^+$, $\mathcal{U}^-$ and $\mathcal{U}^+$ are vectors of the dimension $(N_a - 1)$ containing the limit values y^-, y^+ , u^- and u^+, respectively.

A. Example: optimal cart attack

The solution to problem (10) is applied to the cart system described in Section II. It is assumed that the cart rests at the set-point, i.e. $x(k) = [0 \ \ 0]^T$ at the start of the attack. The goal of the attack is that the cart reaches the maximum distance from the set-point.

The attack horizon is experimentally set to $N_a = 89$ steps which corresponds to a time horizon of 8.9 seconds. It is important to note that this is the minimum time required to achieve the attack goal (a larger horizon does not increase the maximum distance, while a shorter horizon is too short time to reach the maximum). Besides, the effect of the attack after the detection has been computed for a time horizon of $N_e = 55$. In this specific example, the same results would be obtained for an arbitrary time horizon.

Figure 4 shows the solution to the optimization problem and its effects. The solution, quite intuitive, consists of driving the cart to one limit position and then speed it up until the other limit is crossed. At this moment ($t = 8.9$ seconds), an alarm is triggered and the *attack perpetration* phase is finished. The attack is instantly rejected and the backup PID controller takes the control in order to recover the system from the attack. However, the recovery action results in the position $y = 16.74$ at $t = 14.4$ seconds, which poses serious consequences in terms of safety.

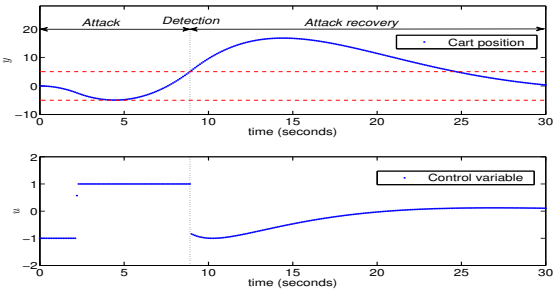

Fig. 4. Optimal cart attack.

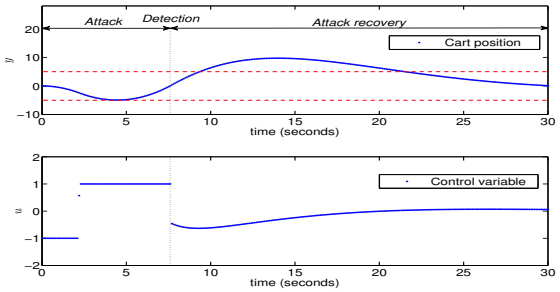

Fig. 5. Cart attack detection.

IV. Attack Detection

The previous attack example in Section III-A demonstrates that detecting attacks by simple limit checking does not provide good results. In this section, a new detection method is presented which does not directly depend on the controller information.

The underlying idea is to figure out, given the current system state, whether limit crossing can be predicted in advance. Let $\mathcal{U}_d$ be a control sequence vector of length N_d, i.e. $\mathcal{U}_d = [u(k), \ldots, u(k+N_d-1)]^T$, and $\mathcal{Y}_d$ be the vector of estimated outputs obtained from $\mathcal{U}_d$, i.e.,

$$\mathcal{Y}_d = \mathcal{G}_x x(k) + \mathcal{G}_u \mathcal{U}_d \tag{11}$$

Where now, with an abuse of notation, matrices $\mathcal{G}_x$ and $\mathcal{G}_u$ are build from an horizon of N_d steps. Then an attack is detected as long as no possible $\mathcal{U}_d \in [\mathcal{U}^- \ \mathcal{U}^+]$ exists such that

$$\mathcal{Y}_d \in [\mathcal{Y}^- \ \ \mathcal{Y}^+] \tag{12}$$

is satisfied for the given current state $x(k)$. This is,

$$\nexists \mathcal{U}_d \in [\mathcal{U}^- \ \ \mathcal{U}^+] : \mathcal{Y}_d \in [\mathcal{Y}^- \ \ \mathcal{Y}^+] \rightarrow \text{attack is detected.}$$

Expression (11) requires the current system state which can be determined by means of state observers. Once the state (or its estimate) is available, the detection procedure turns into a feasibility checking problem. In fact, finding out if a feasible solution exist can be efficiently solved by formulating a linear programing problem with null cost, namely:

$$\max_{\mathcal{U}_d} \quad \begin{bmatrix} 0 & \cdots & 0 \end{bmatrix} \mathcal{U}_d \tag{13a}$$

subject to:

$$\begin{bmatrix} -\mathcal{G}_u \\ \mathcal{G}_u \\ -I \\ I \end{bmatrix} \mathcal{U}_d \leq \begin{bmatrix} -\mathcal{Y}^- + \mathcal{G}_x x(k) \\ \mathcal{Y}^+ - \mathcal{G}_x x(k) \\ -\mathcal{U}^- \\ \mathcal{U}^+ \end{bmatrix} \tag{13b}$$

A. Example: cart attack detection

Consider again the cart system on which the presented detector method will be applied for $N_d = 40$ (again, a larger detection horizon does not improve the results, while with a shorter horizon, the detection could be underestimated). The detector performs the optimization described in (13) and checks whether a feasible solution is possible or not. The results obtained from this new detection method are depicted in Fig. 5.

Note that the attack is now detected at $t = 7.6$ seconds when the cart position is $y = 0.051$ (rather far from the limit) which makes the detection non-intuitive. In summary, the attack is detected at this point by discovering that limit crossing can no longer be avoided owing to the system states (position and speed).

Subsequently, the same recovery strategy as before is considered. Once the attack is detected, the attack is instantly rejected and the PID controller takes over the control of the system. Now, the recovery response is improved by reducing the maximum distance to $y = 9.2$, however, this still does not provide convincing results from a safety perspective.

V. Attack Effect Mitigation

Detecting the attack in advance is not enough. Therefore, advanced steps need to be taken in order to really preserve the system from the attacks. In this section, an optimal control sequence $\mathcal{U}_m$ of length N_m is proposed to mitigate the attack effect and bring the system back to a safety region. In order to commit this goal with the maximum priority no other control performances are sought. Thus, after the control sequence $\mathcal{U}_m$ has accomplished the goal, the backup controller is connected in order to provide suitable control performance as in the nominal (attack-free) case.

Let t_d be the time instant of the attack detection and $x(t_d)$ the corresponding system state at the detection instant. Then the control sequence $\mathcal{U}_m = [u_m(t_d) \ \ldots \ u_m(t_d+N_m-1)]^T$ is computed by minimizing the system output $y(t_d+N_m-1)$ with the condition that limits should not be crossed. However, according to the detectability principle described in Section IV, this minimization problem would become unfeasible since no limit crossing prevention is possible. In this case, soft constraint formulation is used in order to indicate that limit crossing is not desired but allowed. According to this, the intended minimization problem is formulated as

$$\min_{\mathcal{U}_m,\, \xi} \quad y_i(t_d+N_m-1)^T y_i(t_d+N_m-1) + \Gamma \xi^2 \tag{14a}$$

such that:

$$x(k+1) = Ax(k) + Bu_m(k) \tag{14b}$$

$$y(k) = Cx(k) + Du_m(k) \tag{14c}$$

$$\text{for } k = \{t_d, \cdots, t_d + N_m - 1\}$$

$$u_m(t_d), \ldots, u_a(t_d + N_m - 1) \in [u^- \quad u^+] \tag{14d}$$

$$y(t_d), \ldots, y(t_d + N_m - 1) \in [(y^- - \xi) \quad (y^+ + \xi)] \tag{14e}$$

Observe that the variable ξ in (14) is a dummy variable utilized to soften the constraints (14e). Hence, the weight Γ in the cost function (14a) needs to be significantly greater than the other terms.

The minimization problem can be reformulated by taking into account that

$$y(t_d + N_m - 1) = \mathcal{R}x(t_d) + \mathcal{S}\mathcal{U}_m \tag{15}$$

where

$$\mathcal{R} = CA^{N_m - 1}$$

$$\mathcal{S} = \begin{bmatrix} CA^{N_m-2}B & \cdots & CAB & CB & D \end{bmatrix}$$

and again with an abuse of notation, we extend the outputs obtained from $\mathcal{U}_m$ by

$$\mathcal{Y} = \mathcal{G}_x x(t_d) + \mathcal{G}_u \mathcal{U}_m \tag{16}$$

where now matrices $\mathcal{G}_x$ and $\mathcal{G}_u$ are build from an horizon of N_m steps. Finally, the optimization problem in (14) is casted as a quadratic program,

$$\min_{\mathcal{U}_m,\ \xi} \quad \mathcal{U}_m^T \mathcal{S}_i^T \mathcal{S}_i \mathcal{U}_m + 2x(t_d)^T \mathcal{R}_i^T \mathcal{S}_i \mathcal{U}_m + \\ + x(t_d)^T \mathcal{R}_i^T \mathcal{R}_i x(t_d) + \Gamma \xi^2 \tag{17a}$$

subject to:

$$\begin{bmatrix} -\mathcal{G}_u & -\mathbf{1} \\ \mathcal{G}_u & -\mathbf{1} \\ -I & \mathbf{0} \\ I & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathcal{U}_a \\ \xi \end{bmatrix} \leq \begin{bmatrix} -\mathcal{Y}^- + \mathcal{G}_x x(t_d) \\ \mathcal{Y}^+ - \mathcal{G}_x x(t_d) \\ -\mathcal{U}^- \\ \mathcal{U}^+ \end{bmatrix} \tag{17b}$$

where $\mathcal{R}_i$ and $\mathcal{S}_i$ denote the ith-row of the matrices $\mathcal{R}$ and $\mathcal{S}$. Matrix I is an identity matrix of dimension $(N_m - 1)$, while $\mathbf{1}$ and $\mathbf{0}$ denote a column vector of $(N_m - 1)$ ones and zeros. Similar to before, $\mathcal{Y}^-$, $\mathcal{Y}^+$, $\mathcal{U}^-$ and $\mathcal{U}^+$ are vectors of the dimension $(N_m - 1)$ containing the limit values y^-, y^+ , u^- and u^+.

A. Example: cart attack effect mitigation

The optimization problem in (17) is solved for the cart system after the attack is detected at $t_d = 7.6$ seconds, according to Section IV-A. In this case, a time horizon $N_m = 50$ and a weight $\Gamma = 10^6$ are chosen. The obtained results are shown in Fig. 6 where now the maximum distance is $y = 5.68$. The backup PID controller is connected at $t = 12.6$ seconds, just after applying the optimal control sequence $\mathcal{U}_m$ that mitigates the effect of the attack.

Despite the fact that limit crossing is not avoided, the response has significantly improved. As expected from the maximum distance, the value of the dummy variable computed for this example is $\xi = 0.68$, i.e. the exceeded distance. The limit exceedance can be modified by adjusting the weight cost Γ. Remark that there exists an upper bound where increasing Γ does not provide better results. Furthermore large values of Γ do not help on the convergence of the optimization.

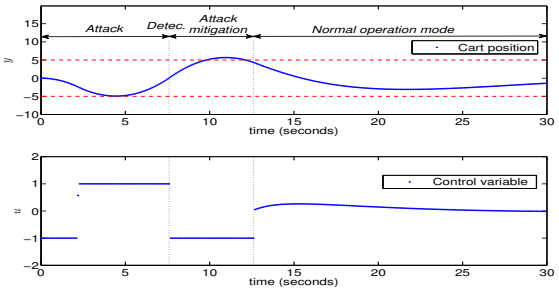

Fig. 6. Cart attack effect mitigation.

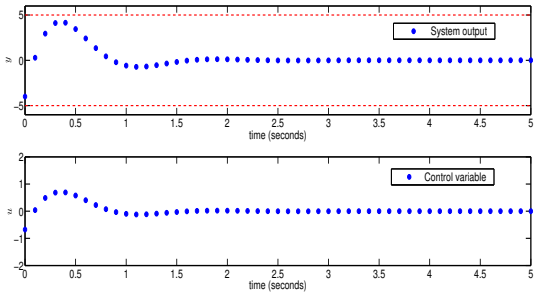

Fig. 7. Typical responses of the system and the controller.

VI. Unstable system example

In order to show the potential of the attack detection and mitigation methods presented in this paper, an unstable system is used in this section as a test bed. The discrete state space representation of the system is

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 1.1 & -0.5 \\ 0.5 & 1.2 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} -1 \\ 2 \end{bmatrix} u(k) \tag{18a}$$

$$y(k) = \begin{bmatrix} -1 & 2 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} \tag{18b}$$

The actuator limits for this example are defined as $[u^- \quad u^+] = [-1.5 \quad 1.5]$ and the measurement safety limits as $[y^- \quad y^+] = [-5 \quad 5]$. The sample time is $T_s = 0.1$ seconds. An LQR controller has been designed in order to stabilize and control the system, where the states of the systems are assumed to be perfectly known. In Fig. 7, typical responses of the attack-free system controlled by the LQR are shown.

An optimal attack has been computed for $N_a = 8$ and $N_e = 1$. Using standard limit checking techniques to detect the attack and protect the system is not feasible. Indeed, as Fig. 8 shows, the attack destabilizes the system and the backup controller is not able to recover it. Since the actuator

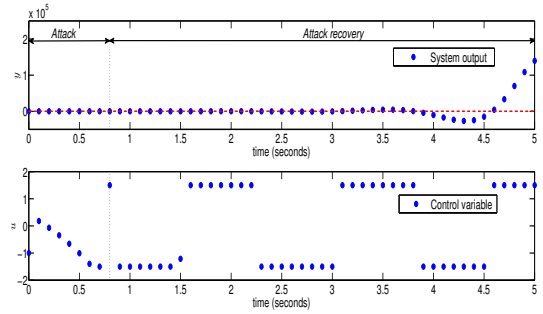

Fig. 8. Optimal attack and detection with standard limit checking detector.

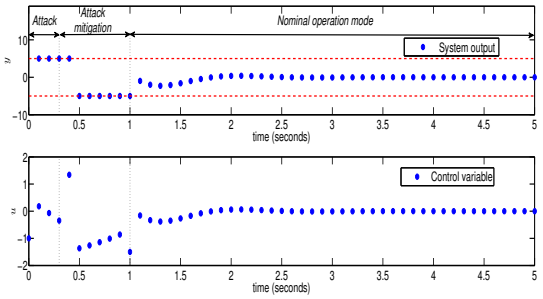

Fig. 9. Attack detection and mitigation with the proposed strategy.

range is limited, the LQR controller can not guarantee stability.

Finally, the proposed methods for attack detection and attack effect mitigation are applied to the same scenario as depicted in Fig. 8. In this case, the detector horizon is $N_d = 10$ and the mitigation horizon $N_m = 7$. The detector triggers an alarm at $t = 0.3$ seconds with enough time to avoid system destabilization in advance. Then, the mitigation action is applied and finally the LQR controller drives the system to the desired state (see Fig. 9).

This example shows that, by applying the proposed methods, not only limit exceeding can be reduced but instability can be prevented which is a major threat in real-word dynamic systems.

VII. Conclusions

The paper shows that relying on standard monitoring systems and backup controllers is not sufficient for system security where planned attacks can be carried out. Therefore, an early but comprehensive study based on optimization techniques has been presented to set up secure control systems. This study involves: worse-case attack analysis, efficient attack detection and mitigation methods.

The present work has been developed from an industrial perspective where the security system does not need to be updated every time the controller is modified. Hence, it is transparent to the control and plant engineers and less vulnerable at the same time.

It is important to point out that the presented optimization problems can be solved efficiently and thus performed online. This is especially true for the detector case where a feasibility problem needs to be solved at each sample time. On the other hand, the mitigation control sequence $\mathcal{U}_m$ is computed at a very critical point when the attack is detected. Note that the performance of the mitigation control sequence is very aggressive since its goal is to reduce the effects of the attack, thus it is not suitable for standard mode operation where other performances may be required.

Although only linear models are used in this paper, the results are satisfactory enough to be considered as basis for future extensions. Further work needs to be done among others, in the methodical selection of the time horizons since the optimality of the solutions strongly depends on it. Uncertainty of the system (e.g. noise, model inaccuracies, communication delays) can also be taken into account as possible extensions, where feedback control loops will be needed.

References

[1] N. Falliere, Murchu, and E. Chien, "W32.stuxnet dossier," Symantec Security Response online report, Symantec, Tech. Rep., February 2011.

[2] J. Slay and M. Miller, *Lessons Learned from the Maroochy Water Breach*, ser. IFIP International Federation for Information Processing. Springer US, 2007, vol. 253, pp. 73–82.

[3] E. Byres and J. Lowe, "The myths and facts behind cyber security risk for industrial control systems," in *In ISA Process Control Conference*, 2003.

[4] A. A. Cárdenas, S. Amin, and S. Sastry, "Research challenges for the security of control systems," in *Proceedings of the 3rd conference on Hot topics in security*, ser. HOTSEC'08. Berkeley, CA, USA: USENIX Association, 2008, pp. 1–6.

[5] ——, "Secure control: Towards survivable cyber-physical systems," in *First International Workshop on Cyber-Physical Systems*, June 2008, pp. 495–500.

[6] S. Amin, A. A. Cárdenas, and S. S. Sastry, "Safe and secure networked control systems under denial-of-service attacks," in *Proceedings of the 12th International Conference on Hybrid Systems: Computation and Control*, ser. HSCC '09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 31–45.

[7] Z.-H. Pang and G.-P. Liu, "Design and implementation of secure networked predictive control systems under deception attacks," *Control Systems Technology, IEEE Transactions on*, vol. 20, no. 5, pp. 1334–1342, September 2012.

[8] 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*, ser. HiCoNS '12. New York, NY, USA: ACM, 2012, pp. 55–64.

[9] I. Shames, A. M. Teixeira, H. Sandberg, and K. H. Johansson, "Distributed fault detection for interconnected second-order systems," *Automatica*, vol. 47, no. 12, pp. 2757 – 2764, 2011.

[10] Y. Mo and B. Sinopoli, "False data injection attacks in control systems," in *First Workshop on Secure Control Systems, Cyber Physical Systems Week 2010*, April 2010.

[11] J. Gertler, *Fault Detection and Diagnosis in Engineering Systems*. New York: Marcel Dekker, Inc., 1998.

[12] M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki, *Diagnosis and Fault-Tolerant Control*, 2nd ed. Springer, 2006.

[13] S. X. Ding, *Model-based Fault Diagnosis Techniques: Design Schemes, Algorithms, and Tools*, 1st ed. Springer Publishing Company, Incorporated, 2008.

[14] R. Isermann, *Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance*. Springer, 2006.

[15] J. M. Maciejowski, *Predictive control with constraints*. Essex, England: Prentice Hall, 2002.

[16] A. Bemporad, M. Morari, V. Dua, and E. Pistikopoulos, "The explicit solution of model predictive control via multiparametric quadratic programming," in *American Control Conference, 2000. Proceedings of the 2000*, vol. 2, 200, pp. 872–876.