4-1-2022

Lyapunov-Based Economic Model Predictive Control for Detecting and Handling Actuator and Simultaneous Sensor/Actuator Cyberattacks on Process Control Systems

Henrique Oyama
Department of Chemical Engineering and Materials Science, Wayne State University, Detroit, MI, hcoyama@wayne.edu

Dominic Messina
Department of Chemical Engineering and Materials Science, Wayne State University, Detroit, MI

Keshav Kasturi Rangan
Department of Chemical Engineering and Materials Science, Wayne State University, Detroit, MI, keshav@wayne.edu

Helen Durand
Department of Chemical Engineering and Materials Science, Wayne State University, Detroit, MI, helen.durand@wayne.edu

Recommended Citation

Oyama H, Messina D, Rangan KK and Durand H (2022) Lyapunov-Based Economic Model Predictive Control for Detecting and Handling Actuator and Simultaneous Sensor/Actuator Cyberattacks on Process Control Systems. Front. Chem. Eng. 4:810129. https://doi.org/10.3389/fceng.2022.810129

frontiers | Frontiers in Chemical Engineering

ORIGINAL RESEARCH
published: 01 April 2022
doi: 10.3389/fceng.2022.810129

Lyapunov-Based Economic Model Predictive Control for Detecting and Handling Actuator and Simultaneous Sensor/Actuator Cyberattacks on Process Control Systems

Henrique Oyama, Dominic Messina, Keshav Kasturi Rangan and Helen Durand**

Department of Chemical Engineering and Materials Science, Wayne State University, Detroit, MI, United States

The controllers for a cyber-physical system may be impacted by sensor measurement cyberattacks, actuator signal cyberattacks, or both types of attacks. Prior work in our group has developed a theory for handling cyberattacks on process sensors. However, sensor and actuator cyberattacks have a different character from one another. Specifically, sensor measurement attacks prevent proper inputs from being applied to the process by manipulating the measurements that the controller receives, so that the control law plays a role in the impact of a given sensor measurement cyberattack on a process. In contrast, actuator signal attacks prevent proper inputs from being applied to a process by bypassing the control law to cause the actuators to apply undesirable control actions. Despite these differences, this manuscript shows that we can extend and combine strategies for handling sensor cyberattacks from our prior work to handle attacks on actuators and to handle cases where sensor and actuator attacks occur at the same time. These strategies for cyberattack-handling and detection are based on the Lyapunov-based economic model predictive control (LEMPC) and nonlinear systems theory. We first review our prior work on sensor measurement cyberattacks, providing several new insights regarding the methods. We then discuss how those methods can be extended to handle attacks on actuator signals and then how the strategies for handling sensor and actuator attacks individually can be combined to produce a strategy that is able to guarantee safety when attacks are not detected, even if both types of attacks are occurring at once. We also demonstrate that the other combinations of the sensor and actuator attack-handling strategies cannot achieve this same effect. Subsequently, we provide a mathematical characterization of the "discoverability" of cyberattacks that enables us to consider the various strategies for cyberattack detection presented in a more general context. We conclude by presenting a reactor example that showcases the aspects of designing LEMPC.

Keywords: cyber-physical system, economic model predictive control, nonlinear systems, cyberattack detection, sensor attack, actuator attack

OPEN ACCESS

Edited by:
Gianvito Vilé,
Politecnico di Milano, Italy

Reviewed by:
Jinfeng Liu,
University of Alberta, Canada
Alexander William Dowling,
University of Notre Dame,
United States

****Correspondence:***
Helen Durand
helen.durand@wayne.edu

Specialty section:
This article was submitted to Computational Methods in Chemical Engineering, a section of the journal Frontiers in Chemical Engineering

***Received:** 06 November 2021*
***Accepted:** 24 January 2022*
***Published:** 01 April 2022*

Citation:
Oyama H, Messina D, Rangan KK and Durand H (2022) Lyapunov-Based Economic Model Predictive Control for Detecting and Handling Actuator and Simultaneous Sensor/Actuator Cyberattacks on Process Control Systems. Front. Chem. Eng. 4:810129. doi: 10.3389/fceng.2022.810129

1 INTRODUCTION

Cyber-physical systems (CPSs) integrate various physical processes with computer and communication infrastructures, which allows enhanced process monitoring and control. Although CPSs open new avenues for advanced manufacturing (Davis et al., 2015) in terms of increased production efficiency, the quality of the production, and cost reduction, this integration also opens these systems to malicious cyberattacks that can exploit vulnerable communication channels between the different layers of the system. In addition to process and network cybersecurity concerns, data collection devices such as sensors and final control elements such as actuators (and signals to or from them) are also potential candidates that can be subject to cyberattacks (Tuptuk and Hailes, 2018). Sophisticated and malicious cyberattacks may affect industrial profits and even pose a threat to the safety of individuals working on site, which motivates attack-handling strategies that are geared toward providing safety assurances for autonomous systems.

There exist multiple points of susceptibility in a CPS framework ranging from communication networks and protocols to sensor measurement and control signal transmission, requiring the development of appropriate control and detection techniques to tackle such cybersecurity challenges (Pasqualetti et al., 2013). To better understand these concerns, vulnerability identification (Ani et al., 2017) has been studied by combining people, process, and technology perspectives. A process engineering-oriented overview of different attack events has been discussed in Setola et al. (2019) to illustrate the impacts on industrial control system platforms. In order to address concerns related to control components, resilient control designs based on state estimates have been proposed for detecting and preventing attacks in works such as Ding et al. (2020) and Cárdenas et al. (2011), wherein the latter cyberattack-resilient control frameworks compare state estimates based on models of the physical process and state measurements to detect cyberattacks. Ye and Luo (2019) address a scenario where actuator faults and cyberattacks on sensors or actuators occur simultaneously by using a control policy based on the Lyapunov theory and adaptation and Nussbaum-type functions.

Cybersecurity-related studies have also been carried out in the context of model-predictive control (MPC; Qin and Badgwell, 2003), an optimization-based control methodology that computes optimal control actions to a process. Specifically, for nonlinear systems, Durand (2018) investigated various MPC techniques with economics-based objective functions [known as economic model predictive controllers (EMPCs) (Ellis et al., 2014a; Rawlings et al., 2012)] when only false sensor measurements are considered. Chen et al. (2020) integrated a neural network-based attack detection approach initially proposed in Wu et al. (2018) with a two-fold control structure, in which the upper layer is a Lyapunov-based MPC designed to ensure closed-loop stability after attacks are flagged. A methodology that may be incorporated as a criterion for EMPC design has been proposed in Narasimhan et al. (2021), in which a control parameter screening based on a residual-based attack detection scheme classifies multiplicative sensor-controller attacks on a process as "detectable," "undetectable," and "potentially detectable" under certain conditions. In addition, a general description of "cyberattack discoverability" (i.e., a certain system's capability to detect attacks) without a rigorous mathematical formalism has been addressed in Oyama et al. (2021).

Prior work in our group has explored the interaction between cyberattack detection strategies, MPC/EMPC design, and stability guarantees. In particular, our prior works have primarily focused on studying and developing control/detection mechanisms for scenarios in which either actuators or sensors are attacked (Oyama and Durand, 2020; Rangan et al., 2021; Oyama et al., 2021; Durand and Wegener, 2020). For example, Oyama and Durand (2020) proposed three cyberattack detection concepts that are integrated with the control framework Lyapunov-based EMPC (Heidarinejad et al., 2012a). Advancing this work, Rangan et al. (2021) and Oyama et al. (2021) proposed ways to consider cyberattack detection strategies and the challenges in cyberattack-handling for nonlinear processes whose dynamics change with time. In the present manuscript, we extend our prior work (which covered sensor measurement cyberattack-handling with control-theoretic guarantees and actuator cyberattack-handling without guarantees) to develop strategies for maintaining safety when actuator attacks are not detected (assuming that no attack occurs on the sensors). These strategies are inspired by the first detection concept in Oyama and Durand (2020) but with a modified implementation strategy to guarantee that even when an undetected actuator attack occurs, the state measurement and actual closed-loop state are maintained inside a safe region of operation throughout the next sampling period.

The primary challenge addressed by this work is the question of how to develop an LEMPC-based strategy for handling sensor and actuator cyberattacks occurring at once. The reason that this is a challenge is that some of the concepts discussed for handling sensor and actuator cyberattacks only work if the other (sensors or actuators) is not under an attack. A major contribution of the present manuscript, therefore, is elucidating which sensor and actuator attack-handling methods can be combined to provide safety in the presence of undetected attacks, even if both undetected sensor and actuator attacks are occurring at the same time. To cast this discussion in a broader framework, we also present a nonlinear systems definition of cyberattack "discoverability," which provides fundamental insights into how attacks can fly under the radar of detection policies. Finally, we elucidate the properties of cyberattack-handling using LEMPC through simulation studies.

The manuscript is organized as follows: following some preliminaries that clarify the class of systems under consideration and the control design (LEMPC) from which the cyberattack detection and handling concepts presented in this work are derived, we review the sensor measurement cyberattack detection and handling policies from Oyama and Durand (2020), which form the basis for the development of the actuator signal cyberattack-handling and combined sensor/actuator cyberattack-handling policies subsequently developed. Subsequently, we propose strategies for detecting and handling cyberattacks on process actuators when the sensor measurements remain intact that are able to maintain safety even when actuator cyberattacks are undetected. We then utilize the insights and developments of the prior sections to clarify which sensor and actuator attack-handling policies can be combined to achieve safety in the presence of combined sensor and actuator cyberattacks. We demonstrate that there are combinations of methods that can guarantee safety in the presence of undetected attacks, even if these attacks occur on both sensors and actuators at the same time (though the other combinations of the discussed methods cannot achieve this). Further insights on the interactions between the detection strategies and control policies for nonlinear systems are presented *via* a fundamental nonlinear systems definition of discoverability. The work is concluded with a reactor study that probes the question of the practicality of the design of control systems that meet the theoretical guarantees for achieving cyberattack-resilience.

2 PRELIMINARIES

2.1 Notation

The Euclidean norm of a vector is indicated by $|\cdot|$, and the transpose of a vector x is denoted by x^T. A continuous function α: $[0, a) \rightarrow [0, \infty)$ is said to be of class $\mathcal{K}$ if it is strictly increasing and $\alpha(0) = 0$. Set subtraction is designated by $x \in A/B := \{x \in R^n : x \in A, x \notin B\}$. Finally, a level set of a positive definite function V is denoted by $\Omega_\rho := \{x \in R^n : V(x) \leq \rho\}$.

2.2 Class of Systems

This work considers the following class of nonlinear systems:

$$\dot{x}(t) = f(x(t), u(t), w(t)) \tag{1}$$

where $x \in X \subset R^n$ and $w \in W \subset R^z$ (W:=$\{w \in R^z \mid |w| \leq \theta_w, \theta_w > 0\}$) are the state and disturbance vectors, respectively. The input vector function $u \in U \subset R^m$, where U:=$\{u \in R^m \mid |u| \leq u^{\max}\}$. f is locally Lipschitz on $X \times U \times W$, and we consider that the "nominal" system of **Eq. 1** ($w \equiv 0$) is stabilizable such that there exist an asymptotically stabilizing feedback control law $h(x)$, a sufficiently smooth Lyapunov function V, and class $\mathcal{K}$ functions $\alpha_i(\cdot)$, $i = 1, 2, 3, 4$, where

$$\alpha_1(|x|) \leq V(x) \leq \alpha_2(|x|) \tag{2a}$$

$$\frac{\partial V(x)}{\partial x} f(x, h(x), 0) \leq -\alpha_3(|x|) \tag{2b}$$

$$\left|\frac{\partial V(x)}{\partial x}\right| \leq \alpha_4(|x|) \tag{2c}$$

$$h(x) \in U \tag{2d}$$

$\forall\, x \in D \subset R^n$ (D is an open neighborhood of the origin). We define $\Omega_\rho \subset D$ to be the stability region of the nominal closed-loop system under the controller $h(x)$ and require that it be chosen such that $x \in X$, $\forall x \in \Omega_\rho$. Furthermore, we consider that $h(x)$ satisfies the following equation:

$$|\hat{h}_i(x) - \hat{h}_i(\hat{x})| \leq L_h|x - \hat{x}| \tag{3}$$

for all $x, \hat{x} \in \Omega_\rho$, with $L_h > 0$, where $\hat{h}_i$ is the i-th component of h.

Since f is locally Lipschitz and $V(x)$ is a sufficiently smooth function, the following holds:

$$|f(x_1, u, w) - f(x_2, u, 0)| \leq L_x|x_1 - x_2| + L_w|w| \tag{4a}$$

$$\left|\frac{\partial V(x_1)}{\partial x} f(x_1, u, w) - \frac{\partial V(x_2)}{\partial x} f(x_2, u, 0)\right| \leq L_x'|x_1 - x_2| + L_w'|w| \tag{4b}$$

$$|f(x_1, u_1, w) - f(x_1, u_2, w)| \leq L_u|u_1 - u_2| \tag{4c}$$

$$|f(x, u, w)| \leq M_f \tag{5}$$

$\forall x_1, x_2 \in \Omega_\rho$, $u, u_1, u_2 \in U$ and $w \in W$, where L_x, L_x', L_w, L_w', L_u, and M_f are positive constants.

We also assume that there are M sets of measurements $y_i \in R^{q_i}$, $i = 1, \ldots, M$, available at t_k as follows:

$$y_i(t) = k_i(x(t)) + v_i(t) \tag{6}$$

where k_i is a vector-valued function, and v_i represents the measurement noise associated with the measurements y_i. We assume that the measurement noise is bounded (i.e., $v_i \in V_i := \{v_i \in R^{q_i} \mid |v_i| \leq \theta_{v,i}, \theta_{v,i} > 0)$ and that measurements of each y_i are continuously available. For each of the M sets of measurements, we assume that there exists a deterministic observer [e.g., a high-gain observer Ahrens and Khalil (2009)] described by the following dynamic equation:

$$\dot{z}_i = F_i(\epsilon_i, z_i, y_i) \tag{7}$$

where z_i is the estimate of the process state from the i-th observer, $i = 1, \ldots, M$, F_i is a vector-valued function, and $\epsilon_i > 0$. When a controller $h(z_i)$ with **Eq. 7** is used to control the closed-loop system of **Eq. 1**, we consider that Assumption 1 and Assumption 2 below hold.

Assumption 1. Ellis et al. (2014b), Lao et al. (2015) There exist positive constants θ_w^*, $\theta_{v,i}^*$, such that for each pair $\{\theta_w, \theta_{v,i}\}$ with $\theta_w \leq \theta_w^*$, $\theta_{v,i} \leq \theta_{v,i}^*$, there exist $0 < \rho_{1,i} < \rho$, $e_{m0i} > 0$ and $\epsilon_{Li}^* > 0$, $\epsilon_{Ui}^* > 0$ such that if $x(0) \in \Omega_{\rho_{1,i}}$, $|z_i(0) - x(0)| \leq e_{m0i}$, and $\epsilon_i \in (\epsilon_{Li}^*, \epsilon_{Ui}^*)$, the trajectories of the closed-loop system are bounded in Ω_ρ, $\forall\, t \geq 0$.

Assumption 2. Ellis et al. (2014b), Lao et al. (2015) There exists $e_{mi}^* > 0$ such that for each $e_{mi} \geq e_{mi}^*$, there exists $t_{bi}(\epsilon_i)$ such that $|z_i(t) - x(t)| \leq e_{mi}$, $\forall\, t \geq t_{bi}(\epsilon_i)$.

3 ECONOMIC MODEL PREDICTIVE CONTROL

EMPC Ellis et al. (2014a) is an optimization-based control design for which the control actions are computed *via* the following optimization problem:

$$\min_{u(t) \in S(\Delta)} \int_{t_k}^{t_{k+N}} L_e(\tilde{x}(\tau), u(\tau))\, d\tau \tag{8a}$$

$$\text{s.t.} \quad \dot{\tilde{x}}(t) = f(\tilde{x}(t), u(t), 0) \tag{8b}$$

$$\tilde{x}(t_k) = x(t_k) \tag{8c}$$

$$\tilde{x}(t) \in X, \forall t \in [t_k, t_{k+N}) \tag{8d}$$

$$u(t) \in U, \ \forall t \in [t_k, t_{k+N}) \tag{8e}$$

where N is called the prediction horizon, and $u(t)$ is a piecewise-constant input trajectory with N pieces, where each piece is held constant for a sampling period with time length Δ. The economics-based stage cost L_e of **Eq. 8a** is evaluated throughout the prediction horizon using the future predictions of the process state $\tilde{x}$ from the model of **Eq. 8b** (the nominal model of **Eq. 1**) initialized from the state measurement at t_k (**Eq. 8c**). The process constraints of **Eq. 8d**, **Eq. 8e** are state and input constraints, respectively. A receding or moving horizon implementation strategy is employed, i.e., the optimization problem is solved every Δ time units (at each sampling time t_k) such that the first of the N pieces of the input vector trajectory that is the optimal solution is applied to the process. The optimal solution at t_k is denoted by $u*(t_i|t_k)$, where $i = k, \ldots, k + N–1$.

Additional constraints that can be added to the formulation in **Eq. 8** to produce a formulation of EMPC that takes advantage of the Lyapunov-based controller $h(\cdot)$, called Lyapunov-based EMPC [LEMPC Heidarinejad et al. (2012a)], are as follows:

$$V(\tilde{x}(t)) \le \rho_e', \quad \forall t \in [t_k, t_{k+N}), \quad \text{if } x(t_k) \in \Omega_{\rho_e'} \tag{9a}$$

$$\frac{\partial V(\tilde{x}(t_k))}{\partial x} f(\tilde{x}(t_k), u(t_k), 0) \le \frac{\partial V(\tilde{x}(t_k))}{\partial x} f(\tilde{x}(t_k), h(\tilde{x}(t_k)), 0), \quad \text{if } \tilde{x}(t_k) \in \Omega_\rho \Big/ \Omega_{\rho_e'} \tag{9b}$$

where $\Omega_{\rho_e'} \subset \Omega_\rho$ is a subset of the stability region that makes Ω_ρ forward invariant under the controller of **Eqs 8–9**.

4 CYBERATTACK DETECTION AND CONTROL STRATEGIES USING LEMPC UNDER SINGLE ATTACK-TYPE SCENARIOS: SENSOR ATTACKS

The major goal of this work is to extend the strategies for LEMPC-based sensor measurement cyberattack detection and handling from Oyama and Durand (2020) to handle actuator attacks and simultaneous sensor measurement and actuator attacks. For the clarity of this discussion, we first review the three cyberattack detection mechanisms from Oyama and Durand (2020).

This section therefore considers a single attack-type scenario (i.e., only the sensor readings are impacted by attacks). The first control/detection strategy proposed in Oyama and Durand (2020) switches between a full-state feedback LEMPC and variations on that control design that are randomly generated over time to probe for cyberattacks by evaluating state trajectories for which it is theoretically known that a Lyapunov function must decrease between subsequent sampling times. The second control/detection strategy also uses full-state feedback LEMPC, but the detection is achieved by evaluating the state predictions based on the current and prior state measurements to flag an attack while maintaining the closed-loop state within a predefined safe region over one sampling period after an undetected attack is applied. The third control/detection strategy was developed using output feedback LEMPC, and the detection is attained by checking among multiple redundant state estimates to flag that an attack is happening when the state estimates do not agree while still ensuring closed-loop stability under sufficient conditions (which include the assumption that at least one of the estimators cannot be affected by the attack). In addition to reviewing the key features of this design, this section will provide several clarifications that were not provided in Oyama and Durand (2020) to enable us to build upon these methods in future sections.

4.1 Control/Detection Strategy 1-S Using LEMPC in the Presence of Sensor Attacks

The control/detection strategy 1-S, which corresponds to the first detection concept proposed in Oyama and Durand (2020), uses full-state feedback LEMPC as the baseline controller and randomly develops other LEMPC formulations with **Eq. 9b** always activated that are used in place of the baseline controller for short periods of time to potentially detect if an attack is occurring. We define specific times at which the switching between the baseline 1-LEMPC and the j-th LEMPC, $j > 1$, happens. Particularly, $t_{s,j}$ is defined as the switching time at which the j-LEMPC is used to drive the closed-loop state to the randomly generated j-th steady-state, and $t_{e,j}$ is the time at which the j-LEMPC switches back to operation under the 1-LEMPC.

The baseline 1-LEMPC is formulated as follows, which is used if $t_{e,j-1} \le t < t_{s,j}$, $j = 2, \ldots$, where $t_{e,1} = 0$:

$$\min_{u_1(t) \in S(\Delta)} \int_{t_k}^{t_{k+N}} L_e(\tilde{x}_1(\tau), u_1(\tau))\, d\tau \tag{10a}$$

$$\text{s.t.} \quad \dot{\tilde{x}}_1(t) = f_1(\tilde{x}_1(t), u_1(t), 0) \tag{10b}$$

$$\tilde{x}_1(t_k) = x_1(t_k) \tag{10c}$$

$$\tilde{x}_1(t) \in X_1, \forall t \in [t_k, t_{k+N}) \tag{10d}$$

$$u_1(t) \in U_1, \ \forall t \in [t_k, t_{k+N}) \tag{10e}$$

$$V_1(\tilde{x}_1(t)) \le \rho_{e,1}', \quad \forall t \in [t_k, t_{k+N}), \ \text{if } \tilde{x}_1(t_k) \in \Omega_{\rho_{e,1}'} \tag{10f}$$

$$\frac{\partial V_1(\tilde{x}_1(t_k))}{\partial x} f_1(\tilde{x}_1(t_k), u_1(t_k), 0) \le \frac{\partial V_1(\tilde{x}_1(t_k))}{\partial x} f_1(\tilde{x}_1(t_k), h_1(\tilde{x}_1(t_k)), 0), \ \text{if } \tilde{x}_1(t_k) \in \Omega_{\rho_1} \Big/ \Omega_{\rho_{e,1}'} \tag{10g}$$

where $x_1(t_k)$ is used, with a slight abuse of the notation, to reflect the state measurement in a deviation variable form from the operating steady state. In addition, in the remainder of this work, f_i ($i \ge 1$) represents the right-hand side of **Eq. 1** when it is written in a deviation variable form from the i-th steady state. u_i represents the input vector in a deviation variable form from the steady-state input associated with the i-th steady state. X_i and U_i correspond to the state and input constraint sets in a deviation variable form from the i-th steady state. In addition, ρ_i and $\rho_{e,i}'$ are associated with the i-th steady state. The addition of a subscript i to the functions in **Eq. 2** (to form h_i, V_i, and $\alpha_{j,i}$, $j = 1, 2, 3, 4$) or M_f also signifies association with the i-th steady state.

The j-th LEMPC, $j > 1$, which is used for $t \in [t_{s,j}, t_{e,j})$, is formulated as follows:

$$\min_{u_j(t)\in S(\Delta)} \int_{t_k}^{t_{k+N}} L_e\left(\tilde{x}_j(\tau), u_j(\tau)\right) d\tau \tag{11a}$$

$$\text{s.t.} \quad \dot{\tilde{x}}_j(t) = f_j\left(\tilde{x}_j(t), u_j(t), 0\right) \tag{11b}$$

$$\tilde{x}_j(t_k) = x_j(t_k) \tag{11c}$$

$$\tilde{x}_j(t) \in X_j, \ \forall\, t \in [t_k, t_{k+N}) \tag{11d}$$

$$u_j(t) \in U_j, \ \forall\, t \in [t_k, t_{k+N}) \tag{11e}$$

$$\frac{\partial V_j\left(\tilde{x}_j(t_k)\right)}{\partial x} f_j\left(\tilde{x}_j(t_k), u_j(t_k), 0\right) \le \frac{\partial V_j\left(\tilde{x}_j(t_k)\right)}{\partial x} \ f_j\left(\tilde{x}_j(t_k), h_j\left(\tilde{x}_j(t_k)\right), 0\right) \tag{11f}$$

where $x_j(t_k)$ represents the state measurement in a deviation variable form from the j-th steady state.

The implementation strategy for this detection method is as follows (the stability region subsets are thoroughly detailed in Oyama and Durand (2020) but reviewed in Remark 1):

1) At a sampling time t_k, the baseline 1-LEMPC receives the state measurement $\tilde{x}_1(t_k)$. Go to Step 2.
2) At t_k, a random number ζ is generated. If this number falls within a range that has been selected to start probing for cyberattacks, randomly generate a j-th steady state, $j > 1$, with a stability region $\Omega_{\rho_j} \subset \Omega_{\rho_{samp2,1}}$ that has a steady-state input within the input bounds, contains the state measurement $\tilde{x}_j(t_k)$, and where $\tilde{x}_j(t_k) \in \Omega_{\rho_{h,j}}/\Omega_{\rho_{s,j}}$. Set $t_{s,j} = t_k$, choose $t_{e,j} = t_{k+1}$, and go to Step 4. Otherwise, if ζ falls in a range that has not been chosen to start probing for cyberattacks or the j-th steady state cannot be generated to meet the conditions above (which include the consideration of the different levels of stability regions), go to Step 3.
3) If $\tilde{x}_1(t_k) \in \Omega_{\rho'_{e,1}}$, go to Step 3a. Else, go to Step 3b.
   a) Compute control signals for the subsequent sampling period with **Eq. 10f** of the 1-LEMPC activated. Go to Step 6.
   b) Compute control signals for the subsequent sampling period with **Eq. 10g** of the 1-LEMPC activated. Go to Step 6.
4) The j-LEMPC receives the state measurement $\tilde{x}_j(t_k)$ and controls the process according to **Eq. 11**. Evaluate the Lyapunov function profile throughout the sampling period. If V_j does not decrease by the end of the sampling period following $t_{s,j}$, or if $\tilde{x}_j(t) \notin \Omega_{\rho_1}$ at any time for $t \in [t_k, t_{k+1})$, detect that the process is potentially under a cyberattack and mitigating actions may be applied. Otherwise, go to Step 5.
5) At $t_{e,j}$, switch back to operation under the baseline 1-LEMPC. Go to Step 6.
6) Go to Step 1 ($k \leftarrow k + 1$).

The first theorem presented in Oyama and Durand (2020) and replicated below guarantees the closed-loop stability of the process of **Eq. 1** under the LEMPCs of **Eqs 10–11** under the implementation strategy described above in the absence of sensor cyberattacks. To follow this and the other theorems that will be presented in this work, the impacts of bounded measurement noise and disturbances on the process state trajectory are characterized in Proposition 1 below, and the bound on the value of the Lyapunov function at different points in the stability region is defined in Proposition 2.

Proposition 1. Ellis et al. (2014b), Lao et al. (2015) Consider the systems below:

$$\dot{x}_i = f_i(x_i(t), u_i(t), w(t)) \tag{12a}$$

$$\dot{\tilde{x}}_i = f_i(\tilde{x}_i(t), u_i(t), 0) \tag{12b}$$

where $|x_i(t_0) - \tilde{x}_i(t_0)| \le \delta$ with $t_0 = 0$. If $x_i(t), \tilde{x}_i(t) \in \Omega_{\rho_i}$ for t ∈ [0, T], then there exists a function $f_{W,i}(\cdot,\cdot)$ such that

$$|x_i(t) - \tilde{x}_i(t)| \le f_{W,i}(\delta, t - t_0) \tag{13}$$

for all $x_i(t), \tilde{x}_i(t) \in \Omega_{\rho_i}$, $u_i \in U_i$, and $w \in W$, with

$$f_{W,i}(s, \tau) := \left(s + \frac{L_{w,i}\theta_w}{L_{x,i}}\right)e^{L_{x,i}\tau} - \frac{L_{w,i}\theta_w}{L_{x,i}} \tag{14}$$

Proposition 2. Ellis et al. (2014b) Let $V_i(\cdot)$ represent the Lyapunov function of the nominal system of **Eq. 1**, in a deviation form from the i-th steady state, under the controller $h_i(\cdot)$ that satisfies **Eqs 2**, **3** for the system of **Eq. 1** when it is in a deviation variable form from the i-th steady state. Then, there exists a function f_{V_i} such that

$$V_i(\bar{x}) \le V_i(\bar{x}') + f_{V_i}(|\bar{x} - \bar{x}'|) \tag{15}$$

$\forall \bar{x}, \bar{x}' \in \Omega_{\rho_i}$ where $f_{V_i}(\cdot)$ is given by

$$f_{V_i}(s) := \alpha_{4,i}\left(\alpha_{1,i}^{-1}(\rho_i)\right)s + M_{V_i}s^2 \tag{16}$$

where M_{V_i} is a positive constant.

Theorem 1. Oyama and Durand (2020) Consider the closed-loop system of **Eq. 1** under the implementation strategy described above and in the absence of a false sensor measurement cyberattack where each controller $h_j(\cdot)$, j ≥ 1, used in each j-LEMPC meets the inequalities in **Eqs 2**, **3** with respect to the j-th dynamic model. Let $\epsilon_{W_j} > 0$, $\Delta > 0$, $N \ge 1$, $\Omega_{\rho_j} \subset \Omega_{\rho_{samp2,1}} \subset \Omega_{\rho_1} \subset X_1$ for j > 1, $\rho_j > \rho_{h,j} > \rho_{\min,j} > \rho_{s,j} > \rho'_{s,j} > 0$, where $\Omega_{\rho_{h,j}}$ is defined as the smallest level set of Ω_{ρ_j} that guarantees that if $V_j(\tilde{x}_j(t_k)) \le \rho_{h,j}$, $V_j(x_j(t_k)) \le \rho_j$, and $\rho_1 > \rho_{samp2,1} > \rho_{samp,1} > \rho'_{e,1} > \rho_{\min,1} > \rho_{s,1} > \rho'_{s,1} > 0$ (where $\Omega_{\rho_{samp,1}}$ is defined as a level set of Ω_{ρ_1} that guarantees that if $x_1(t_k) \in \Omega_{\rho_1}/\Omega_{\rho_{samp,1}}$, then $\tilde{x}_1(t_k) \in \Omega_{\rho_1}/\Omega_{\rho'_{e,1}}$) satisfy

$$-\alpha_{3,j}\left(\alpha_{2,j}^{-1}\left(\rho'_{s,j}\right)\right) + L'_{x,j}M_{f,j}\Delta \le -\epsilon_{w,j}\big/\Delta, \ \ j = 1, 2, \ldots \tag{17}$$

$$\rho'_{e,1} + f_{V,1}\left(f_{W,1}(\delta, \Delta)\right) \le \rho_{samp2,1} \tag{18}$$

$$-\alpha_{3,1}\left(\alpha_{2,1}^{-1}\left(\rho'_{e,1}\right)\right) + L'_{x,1}M_{f,1}\Delta + L'_{x,1}\delta + L'_{w,1}\theta_w \le -\epsilon'_{w,1}\big/\Delta \tag{19}$$

$$-\alpha_{3,j}\left(\alpha_{2,j}^{-1}\left(\rho_{s,j}\right)\right) + L'_{x,j}M_{f,j}\Delta + L'_{x,j}\delta + L'_{w,j}\theta_w \le -\epsilon'_{w,j}\big/\Delta, \quad j = 1, 2, 3, \ldots \tag{20}$$

$$\rho_{\min,j} = \max\left\{V_j\left(x_j(t)\right) : x_j(t_k) \in \Omega_{\rho'_{s,j}}, \ t \in [t_k, t_{k+1}), \ u_j \in U_j\right\}, \quad j = 1, 2, \ldots \tag{21}$$

$$\rho_{samp2,1} \geq \max\left\{V_1(x_1(t)):\ x_1(t_k) \in \Omega_{\rho_{samp,1}} \Big/ \Omega_{\rho'_{e,1}},\right.$$
$$\left. t \in [t_k, t_{k+1}),\ u_1 \in U_1\right\} \tag{22}$$

$$\rho_1 \geq \max\left\{V_1(\tilde{x}_1(t_k)):\ x_1(t_k) \in \Omega_{\rho_{samp2,1}}\right\} \tag{23}$$

$$\rho_j = \max\left\{V_j\big(x_j(t_k)\big):\ \tilde{x}_j(t_k) \in \Omega_{\rho_{h,j}}\right\},\ j = 2, 3, \ldots \tag{24}$$

$$\rho'_{s,j} < \min\left\{V_j\big(x_j(t_k)\big):\ \tilde{x}_j(t_k) \in \Omega_{\rho_j} \Big/ \Omega_{\rho_{s,j}}\right\},\ j = 1, 2, \ldots \tag{25}$$

If $\tilde{x}_1(t_0) \in \Omega_{\rho_{samp2,1}}$, $x_1(t_0) \in \Omega_{\rho_{samp2,1}}$, and $|\tilde{x}_j(t_k) - x_j(t_k)| \leq \delta$, k = 0, 1, . . . , then the closed-loop state is maintained in $\Omega_{\rho_{samp2,1}}$ and the state measurement is in Ω_{ρ_1} when the 1-LEMPC is activated at t_0 and for $t_{e,j-1} \leq t < t_{s,j}$ or when the j-LEMPC is activated for $t_{s,j} \leq t < t_{e,j}$ under the implementation strategy described above, and the closed-loop state and the state measurement are maintained within Ω_{ρ_1} for $t \geq 0$. Furthermore, in the sampling period after $t_{s,j}$, if $\tilde{x}_j(t_k) \in \Omega_{\rho_j}/\Omega_{\rho_{s,j}}$, V_j decreases and $x_j(t) \in \Omega_{\rho_j}$ for $t \in [t_k, t_{k+1})$.

An important clarification regarding the strategy described above that provides more detail compared to (Oyama and Durand, 2020) and aids in understanding the extensions of this method developed later in this work for handling actuator attacks is that the decrease in V_j in Theorem 1 is a decrease in V_j along the closed-loop state trajectory of the actual state (not the measurement). Specifically, that statement in the theorem comes from the following equation in the proof of Theorem 1 in Oyama and Durand (2020), which provides an upper bound on $\dot{V}_j$ along the actual closed-loop state trajectory from t_k to t_{k+1} under an input computed by the *j*-LEMPC when following the implementation strategy described above (i.e., $\tilde{x}_j(t_k) \in \Omega_{\rho_{h,j}}/\Omega_{\rho_{s,j}}$) when **Eq. 20** is satisfied:

$$\frac{\partial V_j\big(x_j(\tau)\big)}{\partial x} f_j\big(x_j(\tau), u_j(t_k), w(\tau)\big) \leq -\alpha_{3,j}\Big(\alpha_{2,j}^{-1}\big(\rho_{s,j}\big)\Big)$$
$$+ L'_{x,j} M_{f,j} \Delta + L'_{x,j}\delta + L'_{w,j}\theta_w \leq -\epsilon'_{w,j} \Big/ \Delta \tag{26}$$

This expression indicates that $V_j(x_j(t)) \leq V_j(x_j(t_0)) - \frac{\epsilon_{w,j}(t-t_0)}{\Delta}$, giving a minimum decrease in V_j of $\epsilon'_{w,j}$ over the sampling period. If this decrease is enough to overcome any measurement noise, such as if

$$\epsilon'_{w,j} > \max_{\tilde{x}_j(t_k) \in \Omega_{\rho_{h,j}} \big/ \Omega_{\rho_{s,j}}} \left| \min\left\{V_j\big(\tilde{x}_j(t_k)\big):\ \tilde{x}_j(t_k) \in \Omega_{\rho_{h,j}} \Big/ \Omega_{\rho_{s,j}}\right\}\right.$$
$$-\max\left\{V_j\big(\tilde{x}_j(t_{k+1})\big):\ \tilde{x}_j(t_k) \in \Omega_{\rho_{h,j}} \Big/ \Omega_{\rho_{s,j}},\right.$$
$$\left.\left. u_j \in U_j,\ |x_j\big(t_p\big) - \tilde{x}_j\big(t_p\big)| \leq \theta_{v,j},\ p = k, k+1\right\}\right| \tag{27}$$

when the input is computed by the *j*-LEMPC (where $\theta_{v,1}$ represents the measurement noise when the full-state feedback is available), then the state measurement must also be decreased by the end of the sampling period. However, at any given time instant, it is not guaranteed to be decreasing due to the noise. An unusual amount of increase could help to flag the attack before a sampling period is over, although this would come from recognizing atypical behavior (essentially pattern recognition).

The reasoning behind the selection of the presented bound on $\epsilon'_{w,j}$ is as follows: the lack of a decrease in the Lyapunov function value between t_k and t_{k+1} is meant to flag an attack. However, with sensor noise, it is possible that **Eq. 26** can hold (which reflects a decrease in the value of V_j evaluated along the trajectory of the actual closed-loop state) but that the decrease in V_j caused by **Eq. 26** is not enough to ensure that V_j evaluated at the measured values of the closed-loop state (instead of the actual values) decreases between t_k and t_{k+1}. For example, consider the case in which the value of V_j barely decreases over a sampling period, so that V_j can be treated as approximately constant. If the noise in the measurements is large, it may then be possible that $V_j(\tilde{x}_j(t_k)) < V_j(\tilde{x}_j(t_{k+1}))$, even though V_j slightly decreased along the actual closed-loop state trajectory (if, for example, the noise originally takes $V_j(\tilde{x}_j(t_k))$ to the minimum possible value, it could be for a given $V_j(x_j(t_k))$, but then at the next sampling time, the Lyapunov function evaluated at the measurement is the maximum possible value that it could take). **Equation 27** ensures that even if this occurs, the decrease in V_j along the actual closed-loop state trajectory is enough to ensure that the maximum value of $V_j(\tilde{x}_j(t_{k+1}))$ is less than the minimum value of $V_j(\tilde{x}_j(t_k))$.

Remark 1. The following relation between the different stability regions has been characterized for Detection Strategy 1-S: $\rho_1 > \rho_{samp2,1} > \rho_{samp,1} > \rho'_{e,1} > \rho_{\min,1} > \rho_{s,1} > \rho'_{s,1} > 0$ (which must hold when the baseline 1-LEMPC is used) and $\rho_j > \rho_{h,j} > \rho_{\min,j} > \rho_{s,j} > \rho'_{s,j} > 0$ for $j > 1$ (which must hold when the *j*-LEMPC is used). The regions $\Omega_{\rho_{samp,1}}$, $\Omega_{\rho_{s,j}}$, and $\Omega_{\rho_{h,j}}$ are important to define due to the presence of measurement noise (Oyama and Durand, 2020). Specifically, Ω_{ρ_j}, j = 1, 2, . . . has been defined as an invariant set in which the closed-loop state is maintained, and $\Omega_{\rho'_{e,1}}$ is a region utilized in distinguishing between whether **Eq. 10f** or **Eq. 10g** is activated in **Eq. 10**. $\Omega_{\rho'_{s,j}}$, j = 1, 2, . . . , is defined as a region such that if the actual state is within $\Omega_{\rho'_{s,j}}$ at a sampling time, the maximum distance that the closed-loop state would be able to go within a sampling period is into $\Omega_{\rho_{\min,j}}$. Furthermore, we define the region $\Omega_{\rho_{s,j}}$ such that if the state measurement is within $\Omega_{\rho_{h,j}}/\Omega_{\rho_{s,j}}$ at t_k, the actual state is outside of $\Omega_{\rho'_{s,j}}$. $\Omega_{\rho_{samp,1}}$ is characterized as a region where, if the actual state is inside this region at a sampling time, the maximum distance that the closed-loop state would be able to travel within a sampling period is into $\Omega_{\rho_{samp2,1}}$. $\Omega_{\rho_{samp2,1}}$ is then defined to be a subset of Ω_{ρ_1} so that the maximum distance that the closed-loop state could go when the state measurement is within $\Omega_{\rho'_{e,1}}$ but the actual state is outside of this region is still inside Ω_{ρ_1}. To ensure that the actual state at t_k is inside Ω_{ρ_j}, we define the region $\Omega_{\rho_{h,j}} \subset \Omega_{\rho_j}$ such that if the state measurement is within $\Omega_{\rho_{h,j}}$ at t_k, the actual state is inside Ω_{ρ_j}.

4.2 Control/Detection Strategy 2-S Using LEMPC in the Presence of Sensor Attacks

The control/detection strategy 2-S, which corresponds to the second detection concept in Oyama and Durand (2020), has been developed using only the 1-LEMPC of **Eq. 10**, and it flags false sensor measurements based on state predictions from the process model from the last state measurement. If the norm of the difference between the state predictions and the current measurements is above a threshold, the measurement is identified as a potential sensor attack. Otherwise, if the norm is below this threshold, even if the measurement was falsified, the closed-loop state can be maintained inside Ω_{ρ_1}, under sufficient conditions Oyama and Durand (2020), for a sampling period after the attack is applied for the process operated under an LEMPC that follows the implementation strategy below, where $\tilde{x}_1(t_k|t_{k-1})$ denotes the prediction of the state $\tilde{x}_1$ at t_k evaluated by integrating the process model of **Eq. 10b** from a measurement at t_{k-1} until t_k:

1) At sampling time t_k, if $|\tilde{x}_1(t_k|t_{k-1}) - \tilde{x}_1(t_k|t_k)| > \nu$, flag that a cyberattack is happening and go to Step 1a. Else, go to Step 1b.
   a) Mitigating actions may be applied (e.g., a backup policy such as the use of redundant controller or an emergency shut-down mode).
   b) Operate the process under the 1-LEMPC of **Eq. 10** while implementing an auxiliary detection mechanism to attempt to flag any undetected attack at t_k. $t_k \leftarrow t_{k+1}$. Go to Step 1.

The second theorem presented in Oyama and Durand (2020), which is replicated below, guarantees the closed-loop stability of the process of **Eq. 1** under the 1-LEMPC of **Eq. 10** under the implementation strategy described above before a sensor attack occurs and for at least one sampling period after the attack.

Theorem 2. Oyama and Durand (2020) Consider the system of **Eq. 1** in closed loop under the implementation strategy described in **Section 4.2** based on a controller $h_1(\cdot)$ that satisfies the assumptions of **Eqs 2**, **3**. Let the conditions of Theorem 1 hold with $t_{s,j} = \infty$, j = 2, 3, . . . , and $\delta \geq f_{W,1}(\theta_{\nu,1}, \Delta) + \nu$. If $\tilde{x}_1(t_0) \in \Omega_{\rho_{samp2,1}} \subset \Omega_{\rho_1}$ and $x_1(t_0) \in \Omega_{\rho_{samp2,1}}$, then $x_1(t) \in \Omega_{\rho_{samp2,1}}$ and the state measurement at each sampling time is in Ω_{ρ_1} for all times before a sampling time t_A that a cyberattack falsifies a state measurement, and $x_1(t) \in \Omega_{\rho_{samp2,1}}$ for $t \in [t_A, t_A + \Delta)$, if the attack is not detected at t_A.

In Theorem 2, δ represents the deviation between the state measurement and the actual state that can be tolerated with the provided closed-loop stability guarantees. If there is no attack, δ corresponds to measurement noise. If there is an attack, then δ reflects the largest possible deviation of the falsified state measurement from the actual state that can be tolerated while the guarantees in the theorem are obtained.

We now provide some additional insights into this strategy compared to Oyama and Durand (2020) in preparation for a discussion about cyberattack "discoverability" later in this work. Specifically, the reason that closed-loop stability can only be guaranteed for a sampling period after an attack in Theorem 2 is due to the use of a state prediction in detecting the attack. Specifically, Theorem 2 ensures that $\tilde{x}_1(t) \in \Omega_{\rho_1}$ and $x_1(t) \in \Omega_{\rho_{samp2,1}}$ for $t < t_A$. According to Oyama and Durand (2020), to demonstrate that $x_1(t) \in \Omega_{\rho_{samp2,1}}$ for $t \in [t_A, t_A + \Delta)$, we consider the measurements $\tilde{x}_1(t_{k-1}|t_{k-1})$ and $\tilde{x}_1(t_k|t_k)$, and the predicted state $\tilde{x}_1(t|t_{k-1})$ from the nominal model of **Eq. 10b** for $t \in [t_{k-1}, t_k]$. Then, as the measurement noise is bounded, $|\tilde{x}_1(t_{k-1}|t_{k-1}) - x_1(t_{k-1})| \leq \theta_{\nu,1}$ and Proposition 1 gives

$$|x_1(t_k) - \tilde{x}_1(t_k|t_{k-1})| \leq f_{W,1}(\theta_{\nu,1}, \Delta) \tag{28}$$

If an attack is not flagged at t_k,

$$\begin{aligned}|x_1(t_k) - \tilde{x}_1(t_k|t_k)| &\leq |x_1(t_k) - \tilde{x}_1(t_k|t_{k-1}) + \tilde{x}_1(t_k|t_{k-1}) - \tilde{x}_1(t_k|t_k)| \\ &\leq f_{W,1}(\theta_{\nu,1}, \Delta) + |\tilde{x}_1(t_k|t_{k-1}) - \tilde{x}_1(t_k|t_k)| \leq f_{W,1}(\theta_{\nu,1}, \Delta) + \nu\end{aligned} \tag{29}$$

We note that **Eqs 28**, **29** assume that there is no attack or an undetected attack at t_{k-1}, respectively, so that $|\tilde{x}_1(t_{k-1}|t_{k-1}) - x_1(t_{k-1})| \leq \theta_{\nu,1}$, which is used in deriving the subsequent requirements on δ that are used to select the parameters of the LEMPC of **Eq. 10** to satisfy Theorem 2. If there is an attack on the sensor measurements at t_{k-1}, it is no longer necessarily true that $|\tilde{x}_1(t_{k-1}|t_{k-1}) - x_1(t_{k-1})| \leq \theta_{\nu,1}$, so that the remainder of the proof would no longer follow. One can see this more explicitly by propagating the bounds in **Eqs 28**, **29**. Specifically, **Eq. 29** allows for the potential that though $|x_1(t_k) - \tilde{x}_1(t_k|t_k)| \leq f_{W,1}(\theta_{\nu,1}) + \nu$, $\tilde{x}_1(t_k|t_k)$ could be falsified. To see the bound on the difference between the state measurement and the actual state that could potentially occur at the next sampling time, we use the fact that $|x_1(t_k) - \tilde{x}_1(t_k|t_k)| \leq \delta$ from **Eq. 29** to derive the following bound like **Eq. 28**:

$$|x_1(t_{k+1}) - \tilde{x}_1(t_{k+1}|t_k)| \leq f_{W,1}(\delta, \Delta) \tag{30}$$

Then, if an attack is not flagged at t_{k+1}, following a procedure similar to that in **Eq. 29** gives

$$|x_1(t_{k+1}) - \tilde{x}_1(t_{k+1}|t_{k+1})| \leq f_{W,1}(\delta, \Delta) + \nu \tag{31}$$

It is reasonable to expect that ν would be set greater than $\theta_{\nu,1}$ since it is reasonable to expect that $|\tilde{x}_1(t_p|t_{p-1}) - \tilde{x}_1(t_p|t_p)|$, $p =$ 0, 1, . . . , could reach values around $\theta_{\nu,1}$ given the bound on the noise; however, whether or not this is the case, the definition of $f_{W,1}$ indicates that the maximum potential difference between the actual state and the (falsified) state measurement is growing with time [i.e., $\theta_{\nu,1} < f_{W,1}(\theta_{\nu,1}, \Delta) + \nu < f_{W,1}(\delta, \Delta) + \nu$]. One could also consider developing δ by performing the analysis of **Eqs 28**, **29**, as is begun in **Eqs 30**, **31**, to obtain a δ that is larger (resulting in greater conservatism in the selection of the LEMPC parameters in Theorem 2 when it is still possible to satisfy the conditions of that theorem with larger values of δ) but that allows multiple sampling periods of the closed-loop state remaining in Ω_{ρ_1} after an attack if desired. Though this is only a maximum bound (i.e., the difference does not necessarily grow in the manner described), this analysis highlights a fundamental difference between measurement noise and disturbances and cyberattacks. Specifically, whereas the conditions of Theorem 2 guarantee recursive feasibility and closed-loop stability in the presence of sufficiently small bounded measurement noise and sufficiently small bounded plant/model mismatch, they cannot make long-term stability guarantees in the presence of false sensor measurements because effectively, those destroy feedback over an extended period of time and leave the process operating in a condition where the inputs being applied are not necessarily tied to the actual or even approximate value of the state (whereas the approximate value of the state may be known from sensor readings in the presence of disturbances and measurement noise). We also highlight that the above discussion can be thought of more generally. For example, one could see how it might become challenging to guarantee resilience against attacks that only slightly offset the measured value of the process state from a predicted value by considering the concept that with noise and disturbances, one would expect that there would be a set of potential initial states that might all be consistent with the noise and disturbance distribution, process model, and measurements. From these initial states, there are potential state trajectories that could all be consistent with the noise and disturbance distribution, process model, and measurements. When feedback is available, it re-restricts the possible range of allowable states from which potentially reasonable final states could be computed once again. In the absence of feedback, the possible final states from the first prediction are then reasonable initial conditions for a second prediction, which, in the presence of noise and disturbances, could potentially significantly expand the number of states that could be consistent with the state. This indicates the mechanism by which an attack could be deceptive.

4.3 Control/Detection Strategy 3-S Using LEMPC in the Presence of Sensor Attacks

The Detection Strategy 3-S, which corresponds to the third detection concept proposed in Oyama and Durand (2020), utilizes multiple redundant state estimators (where we assume that not all of them are impacted by the false sensor measurements) integrated with an output feedback LEMPC and ensures that the closed-loop state is maintained in a safe region of operation for all the times that no attacks are detected. The output feedback LEMPC designed for this detection strategy receives a state estimate z_1 from one of the redundant state estimators (the estimator used to provide state estimates to the LEMPC will be denoted as the i = 1 estimator) at t_k, where the notation follows that of **Eq. 10** with **Eq. 10c** replaced by $\tilde{x}_1(t_k) = z_1(t_k)$ (we will subsequently refer to this LEMPC as the output feedback LEMPC of **Eq. 10**).

This implementation strategy assumes that the process has already been run successfully in the absence of attacks under the output feedback LEMPC of **Eq. 8** for some time such that $|z_i(t) - x(t)| \le \epsilon^*_{mi}$ for all i = 1, . . . , M before an attack:

1) At sampling time t_k, if $|z_i(t_k) – z_j(t_k)| > \epsilon_{\max}$, $i = 1, \ldots, M$, $j = 1, \ldots, M$, or $z_1(t_k) \notin \Omega_\rho$ (where z_1 is the state estimate used in the LEMPC design), flag that a cyberattack is occurring and go to Step 1a. Else, go to Step 1b.
   a) Mitigating actions may be applied (e.g., a backup policy such as the use of redundant controller or an emergency shut-down mode).
   b) Operate using the output feedback LEMPC of **Eq. 10**. $t_k \leftarrow t_{k+1}$. Go to Step 1.

Detection Strategy 3-S guarantees that any cyberattacks that would drive the closed-loop state out of Ω_{ρ_1} will be detected before this occurs. It flags cyberattacks by evaluating the norm of the difference between state estimates. If this norm is above a threshold, which represents "normal" process behavior, the control system is recognized as under a potential sensor attack. To determine a threshold, Oyama and Durand (2020) designed the following bound:

$$
\begin{aligned}
&|z_i(t) - z_j(t)| = |z_i(t) - x(t) + x(t) - z_j(t)| \le |z_i(t) - x(t)| \\
&+|z_j(t) - x(t)| \;\; \le \epsilon_{ij} := \left(e^*_{mi} + e^*_{mj}\right) \le \epsilon_{\max} := \max\left\{\epsilon_{ij}\right\} \qquad (32)
\end{aligned}
$$

for all $i \neq j$, $i = 1, \ldots, M$, $j = 1, \ldots, M$, as long as $t \ge t_q = \max\{t_{b1}, \ldots, t_{bM}\}$. Therefore, abnormal behavior can be detected if $|z_i(t_k) - z_j(t_k)| > \epsilon_{\max}$ if $t_k > t_q$ (this avoids false detections).

The worst-case difference between the state estimate used by the output feedback LEMPC of **Eq. 10** and the actual value of the process state under the implementation strategy above when an attack is not flagged is described in Proposition 3.

Proposition 3. Oyama and Durand (2020) Consider the system of **Eq. 1** under the implementation strategy of **Section 4.3** where $M > 1$ state estimators provide the independent estimates of the process state and at least one of these estimators is not impacted by false state measurements (and the attacks do not begin until after t_q). If a sensor measurement cyberattack is not flagged at t_k according to the implementation strategy, then the worst-case difference between z_i, i ≥ 1, and the actual state $x(t_k)$ is given by

$$
|z_i(t_k) - x(t_k)| \le \epsilon^*_M := \epsilon_{\max} + \max\left\{e^*_{mj}\right\}, \;\; j = 1, \ldots, M \qquad (33)
$$

The third theorem presented in Oyama and Durand (2020), which is replicated below, guarantees the closed-loop stability of the process of **Eq. 1** under the LEMPC of **Eq. 10** under the implementation strategy described above when a sensor cyberattack is not flagged.

Theorem 3. Consider the system of Eq. 1 in a closed loop under the output feedback LEMPC of **Eq. 10** based on an observer and controller pair satisfying Assumption 1 and Assumption 2 and formulated with respect to the i = 1 measurement vector, and formulated with respect to a controller $h(\cdot)$ that meets **Eqs 2**, **3**. Let the conditions of Proposition 3 hold, and $\theta_w \le \theta^*_w$, $\theta_{v,i} \le \theta^*_{v,i}$, $\epsilon_i \in (\epsilon^*_{Li}, \epsilon^*_{Ui})$, and $|z_i(t_0) - x(t_0)| \le e_{m0i}$, for $i = 1, \ldots, M$. Also, let $\epsilon_{W,1} > 0$, $\Delta > 0$, $\Omega_{\rho_1} \subset X$, and $\rho_1 > \rho_{\max} > \rho_{1,1} > \rho'_{e,1} > \rho_{\min,1} > \rho_{s,1} > 0$, satisfy

$$
\begin{aligned}
&\rho'_{e,1} \le \rho_{\max} \\
&\quad - \max\left\{f_V(f_W(\epsilon^*_M, \Delta)), M_f \max\{t_{z1}, \Delta\}\alpha_4(\alpha_1^{-1}(\rho_{\max}))\right\} \qquad (34)
\end{aligned}
$$

$$
\rho'_{e,1} \le \rho_1 - f_V(f_W(\epsilon^*_M, \Delta)) - f_V(\epsilon^*_M) \qquad (35)
$$

$$
-\alpha_3\left(\alpha_2^{-1}\left(\rho_{s,1}\right)\right) + L'_x\left(M_f\Delta + \epsilon^*_M\right) + L'_w\theta_w \le -\epsilon_{W,1}/\Delta \qquad (36)
$$

$$
\rho_{\min,1} = \max\left\{V(x(t)) | V(x(t_k)) \le \rho_{s,1}, \; t \in [t_k, t_{k+1}), \; u \in U\right\} \qquad (37)
$$

$$
\rho_{\min,1} + f_V(f_W(\epsilon^*_M, \Delta)) \le \rho_1 \qquad (38)
$$

$$
\rho_{\max} + f_V(\epsilon^*_M) \le \rho_1 \qquad (39)
$$

where t_{z1} is the first sampling time after t_{b1}, and f_v and f_w are defined as in Proposition 1 and Proposition 2 for i = 1 but with the subscripts dropped. Then, if $x(t_0) \in \Omega_{\rho'_{e,1}}$, $x(t) \in \Omega_{\rho_{\max}}$ for all $t \geq 0$ and $z_1(t_h) \in \Omega_{\rho_1}$ for $t_h \geq \max\{\Delta, t_{z1}\}$ until a cyberattack is detected according to the implementation strategy in **Section 4.3**, if the attack occurs after t_q.

Detection Strategy 3-S does not require the knowledge of which state estimate is false or whether or not it is used by the LEMPC; nevertheless, the proposed approach requires at least one estimator to provide accurate estimates of the actual state so that one of them can check the others (to ensure that there is not a case where all could be consistent but incorrect). As for the other strategies, we conclude with some discussions of this method that provide insights beyond those discussed in Oyama and Durand (2020), here in the form of remarks.

Remark 2. The role of $\Omega_{\rho_{1,1}}$ is to ensure, according to Assumption 1 and Assumption 2, that there exists some time before the closed-loop state, initialized within $\Omega_{\rho_{1,1}}$, leaves Ω_{ρ_1}. Here, $x(t_0) \in \Omega_{\rho'_{e,1}}$, which is taken to be a subset of $\Omega_{\rho_{1,1}}$ for this reason. Specifically, Assumption 1 states that the state of the closed-loop system of **Eq. 1** under inputs computed from the state feedback (with the state feedback not yet meeting the bound in Assumption 2) remains within Ω_{ρ_1} at all times by starting within the interior of Ω_{ρ_1} so that in the time before t_{b1}, the fact that $|z_1 - x(t)| > e_{m1}$ does not cause the closed-loop state of the system of **Eq. 1** to reach the boundary of Ω_{ρ_1} before $|z_1 - x(t)| \leq e_{m1}$, after which point it is assumed that the feedback control law that is stabilizing when it is provided the full-state feedback is receiving state estimates close enough to x to maintain the closed-loop state within Ω_{ρ_1} after t_{b1}. This is true in Theorem 3, where the set in which the closed-loop state is initialized must be sufficiently small such that before t_{b1}, the closed-loop state under the control actions computed by the LEMPC cannot leave Ω_{ρ_1} (even if the state estimates used as the initial condition in the controller are bad). This means, however, that the convergence time t_{b1} for the observer must be sufficiently small to prevent $\rho'_{e,1}$ from needing to be prohibitively small to ensure that the closed-loop state would stay within Ω_{ρ_1} before t_{b1} if it is initialized within $\Omega_{\rho'_{e,1}}$.

Remark 3. Assumption 1 and Assumption 2 are essentially used in Detection Strategy 3-S to imply the existence of observers with convergence time periods that are independent of the control actions applied (i.e., they converge, and stay converged, regardless of the actual control actions applied). High-gain observers are an example of an observer that can meet this assumption (Ahrens and Khalil 2009) for bounded x, u, and w. This is critical to the ability of the multiple observers to remain converged when the process is being controlled by an LEMPC receiving inputs based on the state feedback of only one of them, so that the others are evolving independently of the inputs to the closed-loop system.

Remark 4. We only guarantee in Theorem 3 that $z_1(t) \in \Omega_{\rho_1}$, rather than that $z_j(t) \in \Omega_{\rho_1}$, for all $t \geq 0$ until a cyberattack is detected. This is because $z_1(t) \in \Omega_{\rho_1}$ is required for feasibility of the LEMPC, and the other estimates are not used by the LEMPC and thus they do not impact feasibility. If it was desired to utilize an estimate not impacted by cyberattacks in place of z_1 if an attack on z_1 is discovered, one could develop the parameters of the M possible LEMPCs to meet the requirements of Theorem 3 and then select the operating conditions for the $i = 1$ estimator to be contained in the intersection of the stability regions of all of the others such that any of the other estimators could begin to be used at a sampling time if the $i = 1$ estimator is detected to be compromised at that time. This would require being able to know which of the estimators is not attacked to switch to the correct one when the $i = 1$ estimator is discovered to be attacked.

Remark 5. Larger values of e^*_{mi} (i.e., less accurate state estimates) lead to a larger upper bound ϵ^*_M in Proposition 3, then resulting in a more conservative $\rho'_{e,1}$ according to Theorem 3. This indicates that there is a trade-off between the accuracy of the available state estimators to probe for cyberattacks and the design value of $\rho'_{e,1}$ to ensure closed-loop stability under the proposed output feedback LEMPC cyberattack detection strategy.

Remark 6. The methods for attack detection (Strategies 1-S, 2-S, and 3-S) do not distinguish between sensor faults and cyberattacks. Therefore, they could flag faults as attacks (and therefore, it may be more appropriate to use them as anomaly detection with a subsequent diagnosis step). The benefit, however, is that they provide resilience against attacks if the issue is an attack (which can be designed to be malicious) and not a fault (which may be less likely to occur in a state that an attacker might find particularly attractive). They also flag issues that do not satisfy theoretical safety guarantees, which may make it beneficial to flag the issues regardless of the cause.

5 CYBERATTACK DETECTION AND CONTROL STRATEGIES USING LEMPC UNDER SINGLE ATTACK-TYPE SCENARIOS: ACTUATOR ATTACKS

The methods described above from Oyama and Durand (2020) were developed for handling cyberattacks on process sensor measurements. In such a case, the actuators receive the signals that the controller calculated, but the signal that the controller calculated is not appropriate for the actual process state. This requires the methods to, in a sense, rely on the control actions to show that the sensor measurements are not correct. In contrast, when an attack occurs on the actuator signal, the controller no longer plays a role in which signal the actuators receive. This means that the sensor measurements must be used to show that the control actions are not correct. This difference raises the question of whether the three detection strategies of the prior section can handle actuator attacks or not. This section therefore seeks to address the question of whether it is trivial to utilize the sensor attack-handling techniques from Oyama and Durand (2020) for handling actuator attacks, or if there are further considerations.

We begin by considering the direct extension of all three methods, in which Detection Strategies 1-S, 2-S, and 3-S are utilized in a case where the sensor measurements are intact but the actuators are attacked. In this work, actuator output attacks will be considered to happen when 1) the code in the controller has been attacked and reformulated so that it no longer computes the control action according to an established control law; 2) the control action computed by a controller is replaced by a rogue control signal; or 3) a control action is received by the actuator but subsequently modified at the actuator itself.

When Detection Strategy 1-S is utilized but the actuators are attacked, then at random times, it is intended to utilize the j-LEMPC (however, because of the attack, the control actions from the j-LEMPC are not applied). For an actuator attacker to fly under the radar of the detection strategy, the attacker would need to force a net decrease in V_j along the measured state trajectory between the beginning and end of a sampling period and would need to ensure that the closed-loop state measurement does not leave Ω_{ρ_1} at any point in the sampling period (according to the implementation strategy in **Section 4.1**). This restricts the set of inputs that an attacker can provide in place of those coming from the controller without being detected during a probing maneuver to those that ensure that the closed-loop state does not exit Ω_{ρ_1} throughout the sampling period (ultimately maintaining the closed-loop state within a safe operating region if that region is a superset of Ω_{ρ_1}). Thus, during a probing maneuver, Detection Strategy 1-S, with the flagging of attacks both when V_j along the measurement trajectory does not decrease by the end of a sampling period and when the state measurement leaves Ω_{ρ_1} at any point during a sampling period, provides greater protection from the impacts of attacks on safety when the actuators are attacked than when the sensors are attacked. Specifically, whereas there is no guarantee that an undetected sensor attack would not cause a safety issue when using Detection Strategy 1-S, when an actuator attack occurs instead, then over the sampling period during which a probing maneuver is undertaken, an actuator attacker is unable to cause a safety issue for the closed-loop system without being detected (because the sensor measurements are correct and would flag this problematic behavior before the attacker could cause the closed-loop state to leave a safe operating region). However, because the value of the Lyapunov function at the state measurements is only being checked at the beginning and end of the sampling period, it is possible that the actual closed-loop state could move out of Ω_{ρ_1} over a sampling period when a rogue actuator output is applied, and furthermore that at such a point, the measurement may not show this due to the noise. Therefore, to handle the actuator attacks, it is necessary to add conservatism to the design of the safe operating region compared to Ω_{ρ_1}, so that instead of maintaining the state measurements and closed-loop state within Ω_{ρ_1} only, they are maintained in the supersets of it that prevent the closed-loop state from leaving a safe operating region in the presence of noise and problematic inputs before a sampling period is over. A method for devising such regions is shown in a later section in the context of a combined sensor and actuator attack-handling strategy that makes use of this methodology. If this conservatism is added, then if an actuator attack occurs in a sampling period during which a probing maneuver occurs but it is undetected, the closed-loop state is maintained within the safe operating region. When no probing maneuver is occurring, then if the Lyapunov function evaluated at the state measurement is increasing over a sampling period when the closed-loop state is outside of $\Omega_{\rho'_{e,1}}$, it may be possible that an attack is occurring and that this could be flagged to attempt to catch the attack before the closed-loop state leaves Ω_{ρ_1}; however, as discussed in **Section 4.3**, in the presence of bounded measurement noise, it is possible that V_j may not monotonically decrease when evaluated using the state measurements so that care must be taken in flagging a temporary increase in V_j as a cyberattack to avoid characterizing measurement noise as an attack.

An improved version of Detection Strategy 1-S when there are actuator cyberattacks would only probe constantly for attacks (i.e., the implementation strategy would be the same as that in **Section 4.1**, except that the probing occurs at every sampling time, instead of at random sampling times; this implementation strategy assumes that the regions meeting the requirements in Step 2 in **Section 4.1** can be found at every sampling time, although reviewing when this is possible in detail can be a subject of future work). In this case, since at every sampling time, the attacker would be constrained to choose inputs that cannot cause the state measurement to leave Ω_{ρ_1}, the attacker can never perform an undetected attack that drives the closed-loop state out of a safe operating region before it is detected. This indicates that this modified version of Detection Strategy 1-S (referred to subsequently as Detection Strategy 1-A) is resilient to cyberattacks on actuators in the sense that it is able to prevent an undetected attack from causing safety issues. In light of the question of whether it is trivial to extend Detection Strategy 1-S to handle actuator attacks, we note that Detection Strategy 1-A, which performs continuous probing, is performed in a different manner than Detection Strategy 1-S. Specifically, random probing is used in Detection Strategy 1-S to attempt to surprise an attacker, because the element of surprise is a part of what that algorithm has to counter the fact that the sensor measurements are incorrect. In contrast, Detection Strategy 1-A does not need to have randomized or unpredictable probing; it inherits its closed-loop stability properties from the fact that its design forces the cyberattacker into a corner in terms of what inputs they can apply, even if they fully knew how Detection Strategy 1-A worked, without being detected. This indicates that there is not a 1-to-1 correspondence between how a sensor cyberattack should be handled and how an actuator cyberattack should be handled, with approximately the same strategy. Furthermore, for this strategy, we see a flip in its power between the sensor and actuator attack-handling cases in that Detection Strategy 1-S cannot guarantee safety when a falsified state measurement is provided to the j-LEMPC but can guarantee safety in the presence of an actuator attack during the sampling period after a probing maneuver is initiated if the state measurements are correct.

To further explore how the sensor attack-handling strategies from Oyama and Durand (2020) extend to actuator cyberattack handling, we next consider the use of Detection Strategy 2-S for actuator attacks. This detection strategy is based on state predictions. These predictions must be computed under some inputs, so it is first necessary to consider which inputs these are for the actuator attack extension. Several options for inputs that could be used in making the state predictions include an input computed by a redundant control system, an approximation of the expected control output (potentially obtained *via* fitting the data between state measurements and (non-attacked) controller outputs to a data-driven model), or a signal from the actuator if it is reflective of what was actually implemented. If an actuator signal reflective of the control action that was actually implemented is received and a redundant control system is available, these can be used to cross-check whether the actuator output is correct. This would rapidly catch an attack if the signals are not the same. However, if there is no fully redundant controller (e.g., if actuator signals are available but only an approximation of the expected control output is also available) or if there is a concern that the actuator signals may be spoofed (and there is either a redundant control system or an approximation of the expected control output also available), then state measurements can be used (in the spirit of Detection Strategy 2-S as described in **Section 4.2**) to attempt to handle attacks.

The motivation for considering this latter case in which state measurements and predictions are used to check whether an actuator attack is occurring is as follows: the difference between the redundant control system output or approximation of the control system output and the control output of the LEMPC that is expected to be used to control the process can be checked *a priori*, before the controller is put online. This will result in a known upper bound ϵ_u between control actions that might be computed by the LEMPC and those of the redundant or approximate controller (for the redundant controller, $\epsilon_u = 0$) for a given state measurement. If the state measurements are intact, then the state measurements and predictions under the redundant or approximate controller can be compared to assess the accuracy of the input that was actually applied. The redundant or approximate controller can be used to estimate the input that should be applied to the process, and state predictions can be made using the nominal model of **Eq. 1** to check whether the input that was actually applied to the system seems to be sufficiently similar to the input that was expected (in the sense that it causes the control action that was actually applied to maintain the state measurement in an expected operating region), as it would have under the control action in the absence of an actuator attack, and keeps the norm of the difference between the state prediction and measurement below a bound. Even if $\epsilon_u = 0$, process disturbances and measurement noise could cause the state prediction at the end of a sampling period over which a control action is applied to not fully match the measurement; however, if the error between the prediction and measurement is larger than a bound ν_u that should hold under normal operation considering the noise, value of ϵ_u, and plant/model mismatch, this signifies that there is another source of error in the state predictions beyond what was anticipated, which can be expected to come from the input applied to the process deviating more significantly from what it should have been than was expected (i.e., an actuator attack is flagged). Because the state measurements are correct, the state predictions are always initiated from a reasonably accurate approximation of the closed-loop state; therefore, with sufficient conservatism in the design of Ω_{ρ_1} and a constant monitoring of whether the state measurement leaves that region, the closed-loop state can be prevented from leaving a safe operating region within a sampling period before an attack is detected. We will call the resulting strategy Detection Strategy 2-A. A method for designing a sufficiently conservative control strategy is shown in a later section in the context of a combined sensor and actuator attack-handling strategy that makes use of this methodology. In contrast to Detection Strategy 2-S that can only ensure safe operation for at least one sampling period after a sensor attack is implemented, Detection Strategy 2-A, like Detection Strategy 1-A, can be made fully resilient to actuator cyberattacks in the sense that an undetected attack could not cause safety issues. As long as the actual and predicted inputs are sufficiently close in a norm sense (within ϵ_u of one another), and the disturbances and measurement noise are bounded, then the deviations between the actual and predicted input act as bounded plant/model mismatch (if no attack is detected) that an LEMPC can be designed to handle such that the actual state and predicted state trajectories can still be kept inside a safe region of operation under actuator attacks with the monitoring of whether the state measurement leaves Ω_{ρ_1}. Once again, we see that the modifications to Detection Strategy 2-S, and casting it in a form applicable to actuator attacks rather than sensor attacks, significantly enhances the power of the strategy compared to what can be guaranteed with sensor attacks only.

So far, the extended versions of Detection Strategy 1-S and of 2-S to the actuator-handling case have been more powerful against actuator attacks than Detection Strategies 1-S and 2-S have been against sensor attacks. In contrast, attempting to utilize Detection Strategy 3-S, which enabled safety to be maintained for all times if a sensor measurement attack was undetected (and at least one redundant estimator was not), may result in a strategy that appears to be weaker in the face of actuator attacks. One of the assumptions of Detection Strategy 3-S in **Section 4.3** is that an observer exists that satisfies the conditions in Assumption 1 and Assumption 2. High-gain observers can meet this assumption, and under sufficient conditions, they meet this assumption regardless of the actual value of the input (which was important for achieving the results in Theorem 3 as noted in Remark 3). However, this means that in the case that only the inputs are awry, the state estimates would still be intact because of the convergence assumption, such that they will not deviate from one another in the desired way and Detection Strategy 3-S could not be used as an effective detection strategy for actuator attacks with such estimators. Although a further investigation of whether other types of observer designs or assumptions could be more effective in designing an actuator attack-handling strategy based on Detection Strategy 3-S (to be referred to as Detection Strategy 3-A) could be pursued, these insights again indicate that there are fundamental differences between utilizing the detection strategies for actuator attack-handling compared to sensor attack-handling. The discussion throughout this section therefore seems to suggest that the integrated control and detection frameworks presented above have structures that make them more or less relevant to certain types of attacks and that also affect the extent to which they move toward flexible and lean frameworks with minimal redundancy for cyberattack detection, compared to relying on redundant systems. For example, Detection Strategy 3-S relies on redundant state estimators for detecting sensor attacks, but Detection Strategy 2-A relies on having a redundant controller for detecting actuator attacks. It is interesting in light of this that Detection Strategies 1-A and 1-S do not require redundant control laws but do require many different steady-states to be selected over time. We can also note that the strength of Detection Strategies 1-A and 2-A against actuator attacks above comes partially from the ability of the combined detection and control policies in those cases to set expectations for what the sensor signals should look like that, if not violated, indicate safe operation, and if violated, can flag an attack before safe operation is compromised. As will be discussed later, this has relevance to the notions of cyberattack discoverability in that to cause attacks to be discoverable, integrated detection and control need to be performed such that the control theory can set the expectations for detection to be different if there is an attack or impending safety issue from an attack compared to if not, to force attacks to show themselves. A part of the power of a theory-based control law like Detection Strategy 1-A or 2-A against actuator attacks is the ability to perform that expectation setting.

6 MOTIVATION FOR DETECTION STRATEGIES FOR ACTUATOR AND SENSOR ATTACKS

The above sections addressed how LEMPC might be used for handling sensor attacks or actuator attacks individually. In this section, we utilize a process example to motivate further work on exploring how LEMPC might be used to handle both sensor and actuator attacks. Specifically, consider the nonlinear process model below, which consists of a continuous stirred tank reactor (CSTR) with a second-order, exothermic, irreversible reaction of the form $A \rightarrow B$ with the following dynamics:

$$\dot{C}_A = \frac{F}{V}(C_{A0} - C_A) - k_0 e^{-\frac{E}{R_g T}} C_A^2 \tag{40}$$

$$\dot{T} = \frac{F}{V}(T_0 - T) - \frac{\Delta H k_0}{\rho_L C_p} e^{-\frac{E}{R_g T}} C_A^2 + \frac{Q}{\rho_L C_p V} \tag{41}$$

where the states are the reactant concentration of species A (C_A) and temperature in the reactor (T). The manipulated input is C_{A0} (the reactant feed concentration of species A). The values of the parameters of the CSTR model (F, V, k_0, E, R_g, T_0, ρ_L, ΔH, and C_p) are taken from (Heidarinejad et al., 2012b). The vectors of deviation variables for the states and input from their operating steady-state values, $x_{1s} = [C_{As}\ T_s]^T = [2.00\ \text{kmol/m}^3\ 350.20\ \text{K}]^T$, $C_{A0s} = 4.0$ kmol/m^3, respectively, are $x_1 = [x_{1,1}\ x_{1,2}]^T = [C_A - C_{As}\ T - T_s]^T$ and $u_1 = C_{A0} - C_{A0s}$. The process model represented by **Eqs 40, 41** is numerically integrated using the explicit Euler method with the integration step of 10^{-4} h. The stage cost, for which the time integral is desired to be maximized, is selected to be $L_e = k_0 e^{-E/(R_g T)} C_A^2$. The sampling period was set to $\Delta = 0.01$ h, with the prediction horizon set to $N = 10$. The initial condition for the closed-loop state was 0.7 kmol/m^3 below the steady-state value for C_A and 30 K below the steady-state value for T. The LEMPC simulations were performed using fmincon on a Lenovo model 80XN x64-based ideapad 320 with an Intel(R) Core(TM) i7-7500U CPU at 2.70 GHz, 2,904 Mhz, running Windows 10 Enterprise, in MATLAB R2016b. To ensure that the fmincon solver status was that it stated it had found a local minimum, a variety of initial guesses for the solver were made at a sampling time if it did not find a local minimum using the first guess.

The Lyapunov-based stability constraints in **Eqs 9a**, **9b** were designed using a quadratic Lyapunov function $V_1 = x^T P x$, where $P = [110.11\ 0;\ 0\ 0.12]$. The Lyapunov-based controller utilized was a proportional controller of the form $h_1(x_1) = -1.6x_{1,1} - 0.01x_{1,2}$ (Heidarinejad et al., 2012b) subject to input constraints ($|u_1| \leq 3.5$ kmol/m^3). The stability region was set to $\rho_1 = 440$ (i.e., $\Omega_{\rho_1} = \{x \in R^2\colon V_1(x) \leq \rho_1\}$) and $\rho'_{e,1} = 330$. The LEMPC receives full-state feedback, which is sent to the LEMPC at synchronous time instants t_k. The controller receives a state measurement subject to bounded measurement noise, and the process is subject to bounded disturbances. Specifically, the noise is represented by a standard normal distribution with mean zero, standard deviations of 0.0001 kmol/m^3 and 0.001 K, and bounds of 0.00001 kmol/m^3 and 0.0005 K for the concentration of the reactant and reactor temperatures, respectively. In addition, disturbances were added to the right-hand side of the differential equations describing the rates of change of C_A and T with zero mean and standard deviations of 0.05 kmol/m^3 h and 2 K/h, and bounds of 0.005 kmol/m^3 h and 1 K/h, respectively. Normally distributed random numbers were implemented using the randn function in MATLAB, with a seed of 10 to the random number generator rng.

We first seek to gain insight into the differences between single attack-type cases and simultaneous sensor and actuator attacks. To gain these insights, we will use the strategies inspired by the detection strategies discussed above, but not meeting the theoretical conditions, so that these are not guaranteed to have resilience against any types of attacks (some discussion of moving toward getting theoretical parameters for LEMPC, which elucidates that obtaining the parameters that guarantee cyberattack-resilience for LEMPC formulations in practice should be a subject of future work, will be provided later in this work). Despite the fact that there are no guarantees that any of the strategies used in this example that attempt to detect attacks will do so with the parameters selected, this example still provides a number of fundamental insights into the different characteristics of single attack types compared to simultaneous sensor and actuator attacks, providing motivation for the next results in this work. We also consider that the attack detection mechanisms are put online at the same time as the cyberattack occurs (0.4 h) so that we do not consider that they would have flagged, for example, the changes in the sensor measurements under a sensor measurement attack between the times prior to 0.4 and 0.4 h.

The case studies to be undertaken in moving toward understanding the differences between single and multiple attack-type scenarios involve an LEMPC where the constraint of **Eq. 9b** is enforced at the sampling time, followed by the constraints of the form of **Eq. 9a** enforced at the end of all

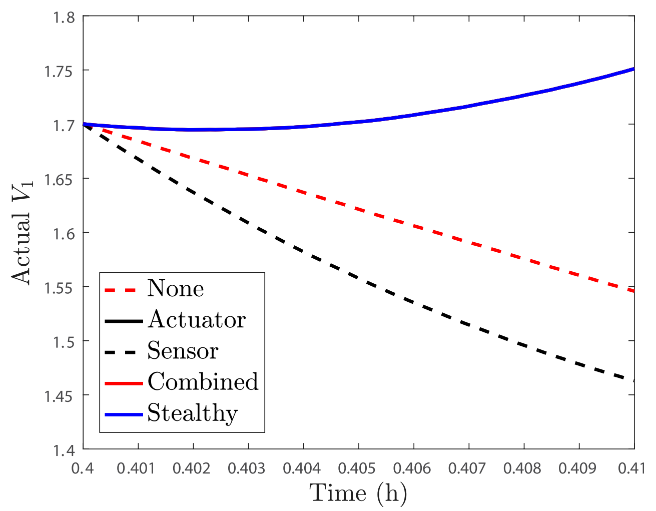

FIGURE 1 | Actual V_1 profiles over one sampling period after 0.4 h of operation for the process example described above in the presence of no attacks ("None"), only actuator cyberattacks ("Actuator"), only sensor attacks ("Sensor"), the baseline combined actuator and sensor attacks ("Combined"), and the stealthy combined sensor and actuator attack ("Stealthy"). The plots for the actuator attack, baseline combined actuator and sensor attack, and stealthy sensor and actuator attack are overlaid due to all having the same input (the false actuator signal) over the sampling period.

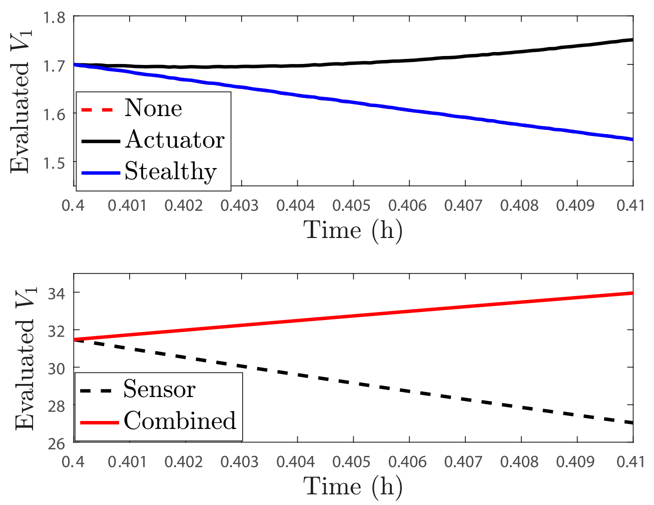

FIGURE 2 | V_1 profiles evaluated using the state measurements over one sampling period after 0.4 h of operation for the process example described above in the presence of no attacks ("None"), only actuator cyberattacks ("Actuator"), only sensor attacks ("Sensor"), the baseline combined actuator and sensor attacks ("Combined"), and the stealthy combined sensor and actuator attack ("Stealthy"). The plots for no attack and for the stealthy combined sensor and actuator attack are overlaid because the stealthy attack provides the no-attack sensor trajectory to the detection device to evade detection.

sampling periods. The first study involves an attack monitoring strategy that involves checking whether the closed-loop state is overall driven toward the origin over a sampling period (if it is not, a possibility of an attack will be flagged). We implement attacks at 0.4 h; sensor attacks are implemented such that the measurement received by the sensor at 0.4 h would be faulty, and an actuator attack would be implemented by replacing the input computed for the time period between 0.4 and 0.41 h with an alternative input. When no attack occurs in the sampling period following 0.4 h of operation, the Lyapunov function evaluated at the actual state and at the state measurement decreases over the subsequent sampling period, as shown in **Figures 1**, **2**.

If instead we consider the case where only a rogue actuator output with the form $u = 0.5$ kmol/m^3 is provided to the process for a sampling period after 0.4 h of operation, **Figures 1**, **2** show that the Lyapunov function profile increases over one sampling period after the attack policy is applied, when the Lyapunov function is evaluated for both the actual state and the measured state, and thus, this single attack event would be flagged by the selected monitoring methodology. Consider now the case where only a false state measurement for reactant concentration, with the form $x_1 + 0.5$ kmol/m^3, is continuously provided to the controller after 0.4 h of operation. This false sensor measurement causes the Lyapunov function value to decrease along the measurement trajectory, as can be seen in **Figure 2**, showing that this attack would not be detected by the strategy. However, it also decreases along the actual closed-loop state trajectory in this case (**Figure 1**) so that no safety issues would occur in this sampling period. This is thus a case when individual attacks would either be flagged over the subsequent sampling period or would not drive the closed-loop state toward the boundary of the safe operating region over that sampling period. Due to the large (order-of-magnitude) difference in the value of V_1 evaluated along the measured state trajectory between the case that the sensor attack is applied and that no attack occurs, as shown in **Figure 1**, it could be argued that this type of attack could be flagged by the steep jump in V_1 between the times prior to the sensor attack that occurs at 0.4 and 0.4 h. However, because we assumed that the method for checking V_1 was not put online until 0.4 h, we assume that it does not have a record of the prior value of V_1 so that we can focus on the trends in this single sampling period after the attacks.

We now consider two scenarios involving the combinations of sensor and actuator attacks. First, we combine the two attacks just described (i.e., false measurements are continuously provided to the controller and detection policies, which have the form $x_1 +$ 0.5 kmol/m^3, and rogue actuator outputs with the form $u =$ 0.5 kmol/m^3 are provided directly to the actuators to replace any inputs computed by the controller). This attack is applied to the process after 0.4 h of operation and subsequently referred to as the "baseline" combined actuator and sensor attack because it is a straightforward extension of the two separate attack policies. In this case, the value of V_1 increases along the measurement trajectory and also increases for the actual closed-loop state so that this attack would be flagged by the proposed policy. In some sense, the addition of the actuator attack made the fact that the system was under some type of attack "more visible" to this detection policy than in the sensor attack-only case (although the individual sensor attack was not causing the closed-loop state to move toward the boundary of the safe operating region so that the lack of detection of an attack in that case would not be considered problematic).

We next consider an alternative combined sensor and actuator attack policy, which we will refer to as a “stealthy” policy. In this case, the attacker provides the exact state trajectory to the detection device that would have been obtained if there was no attack, while at the same time falsifying the inputs to the process. In the case that this same false actuator trajectory was applied to the process and the sensor readings were accurate, we considered that it could be flagged. With the falsified sensor readings occurring at the same time, however, the attack is both undetected and driving the closed-loop state closer to the boundary of the safe operating region over a sampling period. From this, it can be seen that a major challenge arising from combining the attacks is that actuator attack detection policies based on state measurements may fail when attacks are combined, so that the state measurements may imply that the process is operating normally when problematic inputs are being applied.

This raises the question of whether there are alternative detection policies that might flag combined attacks, including those of the stealthy type just described that was “missed” by the detection policy described above where an overall decrease in the Lyapunov function value for the measured state across a sampling period was considered. For example, some of the detection methods described in the prior sections are able to flag actuator attacks before safety issues occur, whereas others flag sensor attacks. This suggests that detection strategies with different strengths might be combined into two-part detection strategies that involve multiple detection methods. To explore the concept of combining multiple methods of attempting to detect attacks (where again this example does not meet theoretical conditions required for resilience and is meant instead to showcase concepts underlying simultaneous attack mechanisms), we consider designing a state estimator for the process to use to compare state estimates against full-state feedback. If the difference between the state estimates and state measurements is larger than a threshold considered to represent abnormal behavior, we will flag that an attack might be occurring. In addition, we will monitor the decrease in the Lyapunov function evaluated along the trajectory of the state measurement over time, and flag a potential attack if it is noticeably increasing across a sampling period.

To implement such a strategy, we must first design a state estimator. We will use the high-gain observer from (Heidarinejad et al., 2012b) with respect to a transformed system state obtained *via* input–output linearization. This estimator (which is redundant because full-state feedback is available) will be used to estimate the reactant concentration of species A from continuously available temperature measurements. The observer equation using the set of new coordinates is as follows:

$$\dot{\hat{z}} = A\hat{z} + L\left(y - C\hat{z}\right) \tag{42}$$

where $\hat{z}$ is the state estimate vector in the new coordinate $z = [x_2\ \dot{x}_2]^T$ Khalil (2002), y is the output measurement, $A = [0\ 1;\ 0\ 0]$, $C = [1\ 0]$, and $L = [100\ 10{,}000]^T$. To obtain the state estimate of the system z, the inverse transformation $T^{-1}(\hat{z})$ is applied.

The next step in designing the detection strategy is to decide on a threshold for the norm of the difference between the state estimate and the state feedback. As a rough attempt to design one that avoids flagging measurement noise and process disturbances as attacks, the data from attack-free scenarios are gathered by simulating the process under different initial conditions and inputs within the input bounds. Particularly, we simulate attack-free events with an end time of 0.4 h of operation for initial conditions in the following discretization: x_1 ranges from -1.5 to $3\ \text{kmol/m}^3$ in the increments of $0.1\ \text{kmol/m}^3$, with x_2 ranging from -50 to 50 K in the increments of 5 K. When these initial conditions are within the stability region, the initial value of the state estimate is found in the transformed coordinates based on the assumption that the initial condition holds. Then, inputs must be generated to apply to the process with noise and disturbances. To explore what the threshold on the difference between the state measurement and estimate might be after 0.4 h to set a threshold to use when the state estimation-based attack detection strategy comes online at that time, we try several different input policies. One is to try $h_1(x)$ at every integration step; if this is done, then the maximum value of the norm of the difference between the state estimate and state measurement at 0.4 h among the scenarios tested is 0.026. If instead a random input policy is used (i.e., at every integration step, a new value of u is generated with mean zero and standard deviation of 2, and bounds on the input of -3.5 and $3.5\ \text{kmol/m}^3$), then the maximum value of the difference between the state estimate and state measurement at 0.4 h among the scenarios tested is 0.122. If instead the random inputs are applied in sample-and-hold with a sampling period of length 0.01 h, the maximum value of the difference between the state estimate and state measurement at 0.4 h is 0.885. If the norm of the error between the state estimate and state measurement is checked at 1 h instead of 0.4 h in the three cases above, the results are 0.003, 0.107, and 0.923, respectively. Though a limited data set was used in these simulations and the theoretical principles of high-gain observer convergence were not reviewed in developing this threshold, 0.923 was selected for the cyberattack detection strategy based on the simulations that had been performed. One could also set the threshold by performing simulations for 0.4 h for a number of different initial states, specifically operated under the LEMPC, instead of the alternative policies above. Changing the threshold in the following discussion could have an impact on attack detection, although there would still be fundamental differences between single attack-type scenarios and simultaneous attack-type scenarios as discussed below.

We next consider the application of the same form of the baseline attacks as described in the prior example occurring at once, i.e., false measurements are continuously provided to the controller, which have the form $x_1 + 0.5\ \text{kmol/m}^3$, and rogue actuator outputs with the form $u = 0.5\ \text{kmol/m}^3$ are provided to the process at 0.4 h of operation. In this combined attack scenario, the norm of the difference between the (falsified) state measurement and the state estimate at 0.4 h is 0.5016, and at 0.41 h is 0.5233, demonstrating that if the threshold is set to a larger number such as 0.923, the state estimate-based detection mechanism does not flag this attack at 0.4 or 0.41 h. **Figure 3** plots the closed-loop state trajectory against the state estimate trajectory over one sampling period after 0.4 h of operation,

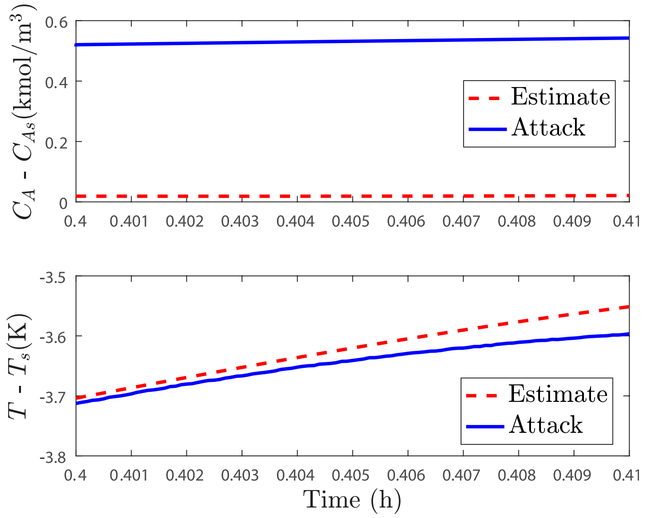

FIGURE 3 | Comparison between the closed-loop state trajectory under attack (solid line) and the closed-loop state estimate trajectory (dashed lines) over one sampling period after 0.4 h of operation under the state feedback LEMPC.

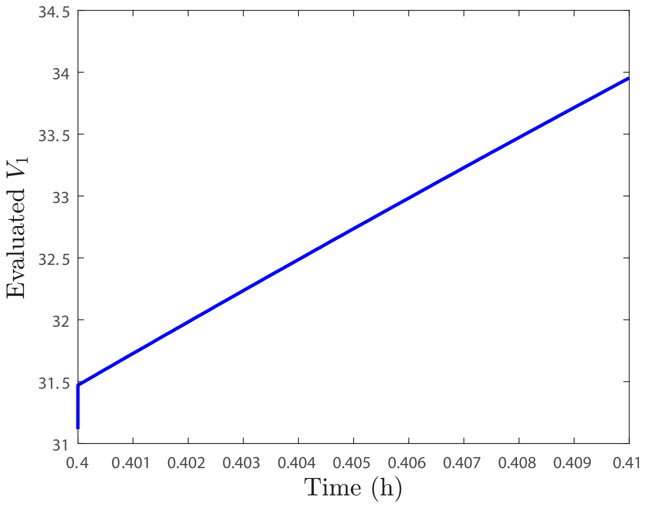

FIGURE 4 | V_1 profile along the measurement trajectory over one sampling period after 0.4 h of operation for the process example in the presence of multiple cyberattack policies (baseline case).

showing the closeness of the trajectories in that time period despite the sensor and actuator attacks at 0.4 h. In fact, if this system is simulated with an actuator attack only, then the difference in the state estimate and state measurement at 0.41 h (the time at which the effects of the actuator attack could first be observed in the process data) is 0.05, showing that with the selected threshold for flagging an attack based on the difference between the state estimate and measurement, the actuator attack only would not be flagged at 0.41 h (despite that there is a net increase in the Lyapunov function value along the measured state trajectory in this case because that is not being checked with only the state estimate-based detection strategy). Considering that the threshold was set based on non-attacked measurements and many different input policies for the threshold set, it is reasonable to expect that an attack would not be flagged if only the input was to change.

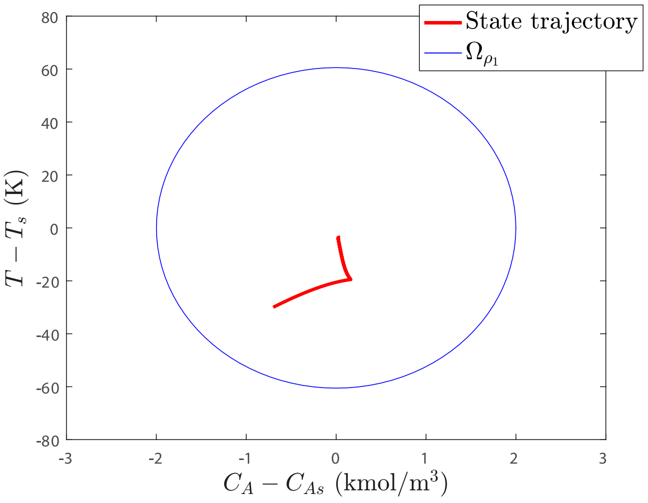

FIGURE 5 | Stability region and closed-loop state trajectory for the process example in the presence of multiple cyberattack policies (baseline case).

For the baseline combined sensor and actuator case, **Figure 4** shows that the Lyapunov function increases over the sampling period after 0.4 h along the measurement trajectory. Therefore, like the case where only the Lyapunov function was checked to attempt to flag this baseline combined attack, the baseline combined sensor and actuator attack can be detected here as well between 0.4 and 0.41 h. Though the attack occurs and is flagged, the closed-loop state was still kept inside the stability region Ω_{ρ_1} over the sampling period that the attacks were applied, as indicated in **Figure 5**.

If the stealthy combined sensor and actuator attack from the prior section is applied, the Lyapunov function value along the closed-loop state trajectory is again increasing, but again, it is decreasing along the estimated trajectory between 0.4 and 0.41 h. However, if this simulation is run longer, then the attack is eventually detected *via* the deviation of the state estimates from the state measurements exceeding the 0.923 threshold, at 0.45 h. In the case that only the Lyapunov function value along the measured state trajectories is checked until 0.45 h, no attack is yet detected, as the Lyapunov function value continues to decrease from 0.4 to 0.45 h along the measured state trajectory. These examples indicate the complexities of having combined sensor and actuator attacks, and also showcase that different detection policies may be better suited for detecting the combined attacks than others. This motivates a further study of the techniques and theory for handling the combined attacks, which is the subject of the next section.

Remark 7. The combined methods illustrated in the examples above do not determine the source of the attacks (e.g., the reason

TABLE 1 | Single attack-type cyberattack detection strategies described.

Detection strategy	Component attacked	Detection/Control policy
Strategy 1–S	Sensor	Random updates to LEMPC
Strategy 2–S	Sensor	Based on state predictions from last state measurement received
Strategy 3–S	Sensor	Based on cross-checks of state estimates between multiple redundant state estimators
Strategy 1–A	Actuator	Updates to LEMPC at every sampling time
Strategy 2–A	Actuator	Based on state predictions under expected inputs
Strategy 3–A	Actuator	Based on cross-checks of state estimates between multiple redundant state estimators

why the Lyapunov function increases could be either due to false sensors, incorrect actuator outputs, or both). However, the nature of a sensor attack differs from a sensor fault. A faulty sensor creates a state trajectory that is not inherently "dynamics based" and intelligently designed to harm a process.

7 INTEGRATED CYBERATTACK DETECTION AND CONTROL STRATEGIES USING LEMPC UNDER MULTIPLE ATTACK TYPE SCENARIOS

The detection concepts described in the prior sections (and summarized in **Table 1**) have been developed to handle only single attack-type scenarios (i.e., either false sensor measurements or rogue actuator signals). However, to make a CPS resilient against different types of cyberattacks, the closed-loop system must be capable of detecting and mitigating scenarios where multiple types of attacks may happen simultaneously. As in the prior sections, detection approaches that not only enable the detection of attacks but that also prevent safety breaches when an attack is undetected are most attractive. This section extends the discussion of the prior sections to ask whether the detection strategies from Oyama and Durand (2020) that were developed for sensor cyberattack-handling and extended to actuator cyberattack-handling above can be used in handling simultaneous sensor and actuator attacks on the control systems.

We first note that based on the discussion in **Section 5**, we do not expect only a single method previously described (Detection Strategies 1-S, 2-S, 3-S, 1-A, 2-A, or 3-A) to be capable of handling both sensor and actuator cyberattacks occurring simultaneously. Instead, to handle the possibility that both types of attacks may occur, we expect that we may need to combine these strategies. However, care must be taken to select and design integrated control/detection strategies such that cyberattack detection and handling are guaranteed even when sensors and/or actuators are under attack. This is because the two types of attacks can interact with one another to degrade the performance of some of the attack detection/handling strategies that work for single attack types as suggested in the example of the prior section. For example, as noted in **Section 5**, in general, sensor measurement cyberattack-handling strategies may make use of correct actuator outputs in identifying attacks, and actuator attack-handling strategies may make use of "correct" (except for the sensor noise) sensor measurements in identifying attacks. If the actuators are no longer providing a correct output, it is then not a given that a sensor measurement cyberattack-handling strategy can continue to be successful, and if the sensors are attacked, it is not a given that an actuator cyberattack-handling strategy can continue to be successful. In this section, we analyze how the various methods in this work perform when these interactions between the sensor and actuator attacks may serve to degrade performance of strategies that worked successfully for only one attack type.

We discuss below the nine possible pairings of actuator and sensor attack-handling strategies based on the detection strategies discussed in this work. The goal of this discussion is to elucidate which of the combined strategies may be successful at preventing simultaneous sensor and actuator attacks from causing safety issues and which could not be based on counterexamples:

- Pairing Detection Strategies 1-S and 1-A: These two strategies essentially have the same construction (where when both are activated, there must be constant changing of the steady states around which the *j*-LEMPCs are designed for constant probing to satisfy the requirements of using Detection Strategy 1-A), in which a decrease in the Lyapunov function value along the measured state trajectories is looked for to detect both the actuator and sensor attacks. Consider a scenario in which an attacker provides sensor measurements that show a decrease in the Lyapunov function value when that would be expected, thus preventing the attack from being detected by the sensors. At the same time, the actuators may be producing inputs unrelated to what the sensors show, which could cause safety issues even if the sensors are not indicating any safety issues, due to attacks occurring on both the sensors and actuators. This pairing is therefore not resilient against combined attacks on the actuators and sensors (i.e., it is not guaranteed to detect attacks that would cause safety issues).
- Pairing Detection Strategies 1-S and 2-A: Detection Strategy 1-S relies on the value of the Lyapunov function decreasing between the beginning and end of a sampling period when the Lyapunov function is evaluated at the state measurement. Detection Strategy 2-A relies on the difference between a state prediction (from the last state measurement and under the expected input corresponding to that measurement) and a state measurement being less than a bound. This design faces a challenge for resilience against simultaneous sensor and actuator attacks in that the

detection strategies for both types of policies depend on the state measurements. Since the state measurements here are falsified, this gives room for any actuator signal to be utilized, and then the sensors to provide readings that suggest that the Lyapunov function is decreasing and that the prediction error is within a bound. Thus, in this strategy, because there is no way to cross-check whether the sensor measurements are correct when there is also an actuator attack, safety is not guaranteed when there are undetected simultaneous attacks.

- Pairing Detection Strategies 1-S and 3-A: Detection Strategy 1-S relies on the state measurement creating a decrease in the Lyapunov function, while Detection Strategy 3-A relies on redundant state estimates being sufficiently close to one another. If Detection Strategy 1-S is not constantly activated (i.e., there is no continuous probing), then because Detection Strategy 3-A may not be guaranteed to detect actuator attacks and Detection Strategy 1-S may not detect them between probing times as described in **Section 5**, this strategy may not be resilient against actuator attacks (and thus also may not be against simultaneous actuator and sensor attacks). However, a slight modification to the strategy to achieve constant probing under Detection Strategy 1-S, forming the pairing of Detection Strategies 3-S and 1-A (because Detection Strategies 3-A and 3-S are equivalent in how they are performed) is resilient against simultaneous sensor and actuator attacks, as is further discussed below. If instead of probing, Ω_{ρ_1} is designed to be a sufficiently conservative subset of a safe operating region, then it could be checked whether at any time, the state measurement leaves Ω_{ρ_1} to flag the attacks; this strategy would also follow similarly to the strategy for detecting attacks using the combination of Detection Strategies 3-S and 1-A for which a proof is provided in a subsequent section.
- Pairing Detection Strategies 2-S and 1-A: This strategy faces similar issues to the combination of Detection Strategies 1-S and 2-A above. Specifically, these strategies again utilize state measurements only to flag attacks, allowing rogue actuator inputs to be applied at the same time as false state measurements without allowing the attacks to be flagged.
- Pairing Detection Strategies 2-S and 2-A: This is a case where only state measurements are being used to flag attacks, so like other methods above where this is insufficient to prevent the masking of rogue actuator trajectories by false sensor measurements, this strategy is also not resilient against attacks.
- Pairing Detection Strategies 2-S and 3-A: Detection Strategy 2-S is based on the expected difference between state predictions and actual states, and Detection Strategy 3-A is based on checking the difference between multiple redundant state estimates. If the threshold for Detection Strategy 2-S is redesigned (forming a pairing that we term as the combination of Detection Strategies 2-A and 3-S below since the threshold redesign must account for actuator attacks as described for Detection Strategy 2-A above to avoid false alarms), the strategy would be resilient to simultaneous actuator and sensor attacks. This is further detailed in the subsequent sections (although it requires that at least one state estimator is not impacted by the attacks).
- Pairing Detection Strategies 3-S and 1-A: This detection strategy can be made resilient against simultaneous actuator and sensor attacks and receives further attention in the following sections to demonstrate and discuss this (though at least one state estimator cannot be impacted by the attacks).
- Pairing Detection Strategies 3-S and 2-A: This strategy can be made resilient for adequate thresholds on the state prediction and state estimate-based detection metrics and will be further detailed below.
- Pairing Detection Strategies 3-S and 3-A: This strategy faces the challenge that it may not enable actuator attacks to be detected because both Detection Strategy 3-S and Detection Strategy 3-A are dependent only on state estimates, which may not reveal incorrect inputs as discussed in **Section 5**. Therefore, it would not be resilient for a case when actuator and sensor attacks could both occur if the redundant observer threshold holds regardless of the applied input.

The above discussion highlights that to handle both the sensor and actuator attacks, a combination strategy cannot be based on sensor measurements alone. In the following sections, we detail how the combination strategies using Detection Strategies 3-S and 1-A, and 3-S and 2-A, can be made resilient against simultaneous sensor and actuator attacks in the sense that, as long as at least one state estimate is not impacted by a false sensor measurement attack, the closed-loop state is always maintained within a safe operating region if attacks are undetected, even if both attack types occur at once. We note that the assumptions that the detectors are intact (e.g., that at least one estimator is not impacted by false sensor measurements or that a state prediction error-based metric is evaluated against its threshold) implies that other information technology (IT)-based defenses at the plant are successful, indicating that the role of these strategies at this stage of development is not in replacing IT-based defenses but in providing extra layers of protection if there are concerns that the attacks could reach the controller itself (while leaving some sensor measurements and detectors uncompromised).

7.1 Simultaneous Sensor and Actuator Attack-Handling *via* Detection Strategies 3-S and 1-A: Formulation and Implementation

In the spirit of the individual strategies Detection Strategy 3-S and Detection Strategy 1-A, a combined policy (to be termed Detection Strategy 1/3) can be developed that uses redundant state estimates to check for sensor attacks (assuming that at least one of the estimates is not impacted by any attack), and also uses different LEMPCs at every sampling time that are designed around different steady-states but contained within a subset of a safe operating region $\Omega_{\rho_{safe}}$ (the subsets are called $\Omega_{\rho_i} \subset \Omega_{\rho_{safe}}$). Under sufficient conditions (which will be clarified in the next section), both the closed-loop state and state estimate are maintained in $\Omega_{\rho_{safe}}$ for all time for the process without attacks or with undetected attacks. The notation to be used for the LEMPC for Detection Strategy 1/3 has the form in **Eq. 11** with **Eq. 11c** replaced by $\tilde{x}_i(t_k) = z_{1,i}(t_k)$ (in this subsection, we will refer to this LEMPC as the *i*-th output feedback LEMPC of **Eq. 11**. The output feedback LEMPC design of **Eq. 11** receives a state estimate $z_{1,i}$ at t_k. In the following, i will be used as a subscript for some of the previously introduced notation to reflect that the quantity is defined for the system in deviation variable form from the *i*-th steady state.

The implementation strategy for Detection Strategy 1/3 assumes that the process has already been run successfully in the absence of attacks under the $i = 1$ output feedback LEMPC of **Eq. 11** for some time (t_q) such that $|z_{j,i}(t) - x_i(t)| \leq \epsilon^*_{mj}$ for all $j = 1, \ldots, M$. In consonance with Oyama and Durand (2020), we consider bounded measurement noise (i.e., $|x_i(t_k) - \tilde{x}_i(t_k)| \leq \theta_{v,i}$). For bounded measurement noise, the subset regions of $\Omega_{\rho_i} \subset \Omega_{\rho_{safe}}$, termed $\Omega_{\rho_{g,i}}$ and $\Omega_{\rho_{h,i}}$, $i \geq 1$, must be considered in the implementation strategy, and they are selected such that if the state measurement is in $\Omega_{\rho_{g,i}} \subset \Omega_{\rho_{h,i}} \subset \Omega_{\rho_1}$, then the closed-loop state and the state measurement are maintained in $\Omega_{\rho_{safe}}$ under sufficient conditions. We assume that no attacks occur before t_q.

1) Before t_q, operate the process under the 1-LEMPC of **Eq. 11**. Go to Step 2.
2) At sampling time t_k, when the *i*-th output feedback LEMPC of **Eq. 11** was just used over the prior sampling period to control the process of **Eq. 1**, if $|z_{j,i}(t_k) - z_{p,i}(t_k)| > \epsilon_{\max}$, $j = 1, \ldots, M$, $p = 1, \ldots, M$, or $\tilde{x}_i(t_k) = z_{1,i}(t_k) \notin \Omega_{\rho_i} \subset \Omega_{\rho_{safe}}$, detect that a cyberattack is occurring and go to Step 3. Else, go to Step 4 ($i \leftarrow i + 1$).
3) Mitigating actions may be applied (e.g., a backup policy such as the use of a redundant controller or an emergency shutdown mode).
4) Select a new *i*-th steady-state. This steady-state must be such that the closed-loop state measurement in deviation form from the new steady-state $\tilde{x}_i(t_k)$ is not in a neighborhood $\Omega_{\rho_{s,i}}$ of the *i*-th steady-state. This steady-state must be such that $\tilde{x}_i(t_k) \in \Omega_{\rho_{g,i}} \subset \Omega_{\rho_{h,i}} \subset \Omega_{\rho_i} \subset \Omega_{\rho_1} \subset \Omega_{\rho_{safe}}$ and the steady-state input is within the input bounds ($\Omega_{\rho_{g,i}}$ is selected such that if the state measurement at t_k is in $\Omega_{\rho_{g,i}}$ then the closed-loop state and the state estimate are maintained in $\Omega_{\rho_i} \subset \Omega_{\rho_{safe}}$ over the subsequent sampling period under sufficient conditions). Go to Step 5.
5) The control actions computed by the *i*-LEMPC of **Eq. 11** for the sampling period from t_k to t_{k+1} is used to control the process according to **Eq. 11**. Go to Step 6.
6) Evaluate the Lyapunov function at the beginning and end of the sampling period, using the state measurements. If V_i does not decrease over the sampling period or if $\tilde{x}_i(t_{k+1}) = z_{1,i}(t_{k+1})$ is not within $\Omega_{\rho_i} \subset \Omega_{\rho_{safe}}$ or Ω_{ρ_1}, detect that the process is potentially under a cyberattack. Go to Step 3. Else, go to Step 7.
7) ($t_k \leftarrow t_{k+1}$). Go to Step 2.

Remark 8. Though the focus of the discussions has been on preventing safety issues, it is possible that the detection and control policies described in this work may sometimes detect other types of malicious attacks that attempt to spoil products or cause a process to operate inefficiently to attack economics. The impacts of the probing strategies on process profitability (compared to routine operation) can be a subject of future work.

7.1.1 Simultaneous Sensor and Actuator Attack-Handling *via* Detection Strategies 3-S and 1-A: Stability and Feasibility Analysis

In this section, we prove recursive feasibility and safety of the process of **Eq. 1** under the LEMPC formulations of the output feedback LEMPCs of **Eq. 11** whenever no sensor or actuator attacks are detected according to the implementation strategy in **Section 7.1** in the presence of bounded measurement noise. The theorem below characterizes the safety guarantees (defined as maintaining the closed-loop state in $\Omega_{\rho_{safe}}$) of the process of **Eq. 1** for all time under the implementation strategy of **Section 7.1** when no sensor and actuator cyberattacks are detected.

Theorem 4. Consider the closed-loop system of **Eq. 1** under the implementation strategy of **Section 7.1** (which assumes the existence of a series of steady-states that can satisfy the requirements in Step 4), where the switching of the controllers at sampling times starts after t_q and no sensor or actuator cyberattack is detected with the i-th output feedback LEMPC of **Eq. 11** based on an observer and controller pair satisfying Assumption 1 and Assumption 2 (in which at least one of the state estimators is not affected by false state measurements) and formulated with respect to the j = 1 measurement vector, and where each controller $h_i(\cdot)$, i ≥ 1, used in each i-LEMPC meets the inequalities in **Eqs 2**, **3** with respect to the i-th dynamic model. Let $\theta_{w,i} \leq \theta^*_{w,i}$, $\theta_{v,i} \leq \theta^*_{v,i}$, $\epsilon_i \in (\epsilon^*_{Li}, \epsilon^*_{Ui})$, and $|z_{j,i}(t_0) - x_i(t_0)| \leq e_{m0j,i}$, for $j = 1, \ldots, M$. Let $\epsilon_{W,i} > 0$, $\Delta > 0$, $N \geq 1$, $\rho_{safe} > \rho_{samp4} > \rho_{samp3} > \rho_1 > \rho_{h,1} > \rho'_{h,1}$, $\Omega_{\rho_p} \subset \Omega_{\rho_1} \subset X_1$ for $P \geq 2$, $\rho_i > \rho_{h,i} > \rho_{g,i} > \rho_{\min,i} > \rho_{s,i} > \rho'_{s,i} > 0$, where $\Omega_{\rho_{g,i}}$ is defined as a level set within $\Omega_{\rho_i} \subset \Omega_{\rho_1} \subset \Omega_{\rho_{safe}}$ that guarantees that if $V_i(z_{1,i}(t_k)) \leq \rho_{g,i}$, $V_i(x_i(t_k)) \leq \rho_{h,i}$. Let the following inequalities be satisfied:

$$\rho_{g,i} = \max\left\{V_i(z_{1,i}(t_k)) \colon V_i(x_i(t_k)) \leq \rho_{h,i},\ i = 2, \ldots, \right. \\ \left. |z_{1,i}(t_k) - x_i(t_k)| \leq \epsilon^*_{M,i}\right\} \tag{43}$$

$$\rho'_{h,1} \leq \rho_{h,1} - M_{f,1} \max\{t_{z1}, \Delta\} \alpha_{4,1}\left(\alpha^{-1}_{1,1}(\rho_1)\right) \tag{44}$$

$$-\alpha_{3,i}\left(\alpha^{-1}_{2,i}\left(\rho'_{s,i}\right)\right) + L'_{x,i}\left(M_{f,i}\Delta + \epsilon^*_{M,i}\right) + L'_{w,i}\theta_{w,i} \leq -\epsilon_{W,i}/\Delta, \\ i = 1, 2, \ldots \tag{45}$$

$$\rho_{h,i} + f_{V,i}\left(\epsilon^*_{M,i}\right) < \rho_i,\ i = 1, 2, \ldots \tag{46}$$

$$\rho_{\min,i} = \max\left\{V_i(x_i(t)) \colon x_i(t_k) \in \Omega_{\rho'_{s,i}}\right\},\ t \in [t_k, t_{k+1}),\ u_i \in U_i, \\ i = 1, 2, \ldots \tag{47}$$

$$\rho'_{s,i} < \min\left\{V_i(x_i(t_k)) \colon z_{1,i}(t_k) \in \Omega_{\rho_{g,i}}/\Omega_{\rho_{s,i}},\ |z_{1,i}(t_k) - x_i(t_k)| \leq \epsilon^*_{M,i}\right\}, \\ i = 1, 2, \ldots \tag{48}$$

$$\epsilon_{W,i} > \max_{z_{1,i}(t_k) \in \Omega_{\rho_{g,i}}/\Omega_{\rho_{s,i}}} \left| \min\left\{V_i(z_{1,i}(t_k)) \colon z_{1,i}(t_k) \in \Omega_{\rho_{g,i}}/\Omega_{\rho_{s,i}}\right\} \right. \\ -\max\left\{V_i(z_{1,i}(t_{k+1})) \colon z_{1,i}(t_k) \in \Omega_{\rho_{g,i}}/\Omega_{\rho_{s,i}},\ u_i \in U_i, \right. \\ \left. \left. |x_i(t_p) - z_{1,i}(t_p)| \leq \epsilon^*_{M,i},\ p = k, k+1\right\} \right| \tag{49}$$

$$\rho_{samp3} = \max\Big\{V_i(x_i(t_k)) : z_{1,i}(t_k) \in \Omega_{\rho_1},\ i = 2, \ldots, \\ |z_{1,i}(t_k) - x_i(t_k)| \le \epsilon^*_{M,i}\} \tag{50}$$

$$\rho_{samp4} = \max\Big\{V_i(x_i(t)) : V_i(x_i(t_k)) \le \rho_{samp3}, \\ i = 2, \ldots,\ u_i(t_k) \in U_i,\ t \in [t_k, t_{k+1})\} \tag{51}$$

where t_{z1} is the first sampling time after t_{b1}, $i = 1, \ldots, M$. Then, if $x_1(t_0) \in \Omega_{\rho'_{h,1}}$, $x_i(t) \in \Omega_{\rho_{safe}}$ for all $t \ge 0$ and $z_{1,i}(t) \in \Omega_{\rho_{safe}}$ for $t \ge \max\{\Delta, t_{z1}\}$ until a cyberattack is detected according to the implementation strategy in **Section 7.1**, if the attack occurs after t_q under the i-th LEMPC.

Proof 1. The proof consists of four parts. In Part 1, the feasibility of the i-th output feedback LEMPC of **Eq. 11** is proven when $z_{1,i}(t_k) \in \Omega_{\rho_i}$. In Part 2, we show that the closed-loop state trajectory is contained in $\Omega_{\rho_{h,1}} \subset \Omega_{\rho_{safe}}$ for $t \in [t_0, \max\{\Delta, t_{z1}\})$. In Part 3, we prove that for $t > \max\{\Delta, t_{z1}\}$ but before an attack occurs, $x_i(t)$ and $z_{1,i}(t)$ are bounded within Ω_{ρ_1}, and that $(V_i(t_{k+1}) - V_i(t_k)) < 0$. In Part 4, we prove that if there is an attack (either a false sensor measurement attack, actuator attack, or both) at t_k but it is not detected using the proposed control/detection strategy (i.e., $|z_{j,i}(t) - z_{p,i}(t)| \le \epsilon_{\max_i}$ and $(V_i(t_{k+1}) - V_i(t_k)) < 0$, for all $j = 1, \ldots, M$, $p = 1, \ldots, M$), $x_i(t)$ and $z_{1,i}(t)$ are bounded in $\Omega_{\rho_{safe}}$.

Part 1. The Lyapunov-based controller h_i implemented in sample-and-hold is a feasible solution to the i-th output feedback LEMPC of **Eq. 11** when $\tilde{x}_i(t_k) = z_{1,i}(t_k) \in \Omega_{\rho_i} \subset \Omega_{\rho_{safe}}$. Specifically, $h_i(\tilde{x}(t_p))$, $p = k, \ldots, k + N-1$, $t \in [t_p, t_{p+1})$, is a feasible solution to the i-th output feedback LEMPC of **Eq. 11** because it meets the input constraints of **Eq. 11e** according to **Eq. 2**, it trivially satisfies **Eq. 11f**, and it satisfies **Eq. 11d** when $\tilde{x}_i(t) \in \Omega_{\rho_i} \subset X_i$ according to the implementation strategy in **Section 7.1**. $h_i(\tilde{x}(t_p))$, $p = k, \ldots, k + N - 1$, $t \in [t_p, t_{p+1})$, ensures that $\tilde{x}_i(t) \in \Omega_{\rho_i}$ by the properties of the Lyapunov-based controller Muñoz de la Peña and Christofides (2008) where, if the conditions of **Eqs 45**, **47** are met, then if $\tilde{x}_i(t_p) \in \Omega_{\rho_i}/\Omega_{\rho'_{s,i}}$, $V_i(\tilde{x}_i)$ decreases throughout the following sampling period (keeping the closed-loop state in Ω_{ρ_i}), or if $\tilde{x}_i(t_p) \in \Omega_{\rho'_{s,i}}$, $\tilde{x}_i(t) \in \Omega_{\rho_{\min,i}} \subset \Omega_{\rho_i}$ for $t \in [t_p, t_{p+1})$.

Part 2. To demonstrate boundedness of the closed-loop state in $\Omega_{\rho_1} \subset \Omega_{\rho_{safe}}$ for $t \in [t_0, \max\{\Delta, t_{z1}\})$, the Lyapunov function along the closed-loop state trajectory can be evaluated as follows:

$$V_1(x_1(t)) = V_1(x_1(t_0)) + \int_{t_0}^{t} \frac{\partial V_1(x_1(\tau))}{\partial \tau} d\tau = V_1(x_1(t_0)) \\ + \int_{t_0}^{t} \frac{\partial V_1(x_1(\tau))}{\partial x} \dot{x}_1(\tau) d\tau \quad \le \rho'_{h,1} + M_{f,1} \max\{\Delta, t_{z1}\} \alpha_{4,1}\left(\alpha_{1,1}^{-1}(\rho_1)\right) \tag{52}$$

for all $t \in [t_0, \max\{\Delta, t_{z1}\})$, where the latter inequality follows from **Eqs 2**, **5**, and $x(t_0) \in \Omega_{\rho'_{h,1}} \subset \Omega_{\rho_1}$. If $\rho'_{h,1}$ satisfies **Eq. 44**, then $V_1(x_1(t)) \le \rho_{h,1}, \forall t \in [t_0, \max\{\Delta, t_{z1}\})$, i.e., $x_1(t) \in \Omega_{\rho_{h,1}} \subset \Omega_{\rho_1}$ for all $t \in [t_0, \max\{\Delta, t_{z1}\})$. The state estimate is also maintained within Ω_{ρ_1} at t_{z1} if **Eq. 46** and Proposition 2 hold and there is no attack, because then,

$$V_1\left(z_{1,1}(t_{z1})\right) \le V_1(x_1(t_{z1})) + f_{V,1}\left(|z_{1,1}(t_{z1}) - x_1(t_{z1})|\right) \le \rho_{h,1} \\ + f_{V,1}\left(\epsilon^*_{M,1}\right) < \rho_1 \tag{53}$$

Part 3. To demonstrate the boundedness of the closed-loop state and state estimate in $\Omega_{\rho_{safe}}$ for $t \ge [t_0, \max\{\Delta, t_{z1}\})$, we first consider that the process is not experiencing a cyberattack (i.e., $|z_{j,i}(t_k) - x_i(t_k)| \le \max(e_{mj,i})$, for all $j = 1, \ldots, M$). Since $x_1(t_{z1}) \in \Omega_{\rho_{h,1}} \subset \Omega_{\rho_1}$ and $z_{1,1}(t_{z1}) \in \Omega_{\rho_1}$ from Part 1, the implementation strategy of **Section 7.1** can be executed at t_{z1}, and according to Step 4, $x_i(t_{z1})$ will be contained in $\Omega_{\rho_{h,i}}$. Similar to the steps presented in the third theorem in Oyama and Durand (2020), considering **Eqs 11f**, **Eq. 2**, **4b**, the bound on w_i, and adding and subtracting the term $\frac{\partial V_i(\tilde{x}_i(t_k))}{\partial x} f_i(\tilde{x}_i(t_k), u_i(t_k), 0)$ to/from $\dot{V}_i(x_i(t)) = \frac{\partial V_i(x_i(t))}{\partial x} f_i(x_i(t), u_i(t_k), w_i(t))$ and using the triangle inequality, we obtain

$$\dot{V}_i(x_i(t)) \le -\alpha_{3,i}(|\tilde{x}_i(t_k)|) + L'_{x,i}|x_i(t) - \tilde{x}_i(t_k)| + L'_{w,i}\theta_{w,i} \tag{54}$$

From $|x_i(t) - \tilde{x}_i(t_k)| \le |x_i(t) - x_i(t_k)| + |x_i(t_k) - \tilde{x}_i(t_k)|$, and from **Eq. 11c** with $\tilde{x}_i(t_k) = z_{1,i}(t_k)$, we obtain that:

$$|x_i(t) - \tilde{x}_i(t_k)| \le |x_i(t) - x_i(t_k)| + \epsilon^*_{M,i} \tag{55}$$

From **Eqs 5**, **54**, **55**, and considering $\tilde{x}_i(t_k) \in \Omega_{\rho_{g,i}}/\Omega_{\rho_{s,i}}$:

$$\dot{V}_i(x_i(t)) \le -\alpha_{3,i}\left(\alpha_{2,i}^{-1}\left(\rho_{s,i}\right)\right) + L'_{x,i}\left(M_{f,i}\Delta + \epsilon^*_{M,i}\right) + L'_{w,i}\theta_{w,i} \tag{56}$$

for all $t \in [t_k, t_{k+1})$. According to the implementation strategy in **Section 7.1**, when $z_{1,i}(t_k) \in \Omega_{\rho_{g,i}}/\Omega_{\rho_{s,i}}$, then $x_i(t_k) \in \Omega_{\rho_{h,i}}/\Omega_{\rho'_{s,i}}$ by **Eqs 43**, **48**. If the condition of **Eq. 45** is satisfied, **Eq. 56** gives:

$$V_i(x_i(t)) \le V_i(x_i(t_k)) - \frac{\epsilon_{W,i}(t - t_k)}{\Delta},\ t \in [t_k, t_{k+1}) \tag{57}$$

Thus, when $x_i(t_k) \in \Omega_{\rho_{h,i}}/\Omega_{\rho'_{s,i}}$ and $z_{1,i}(t_k) \in \Omega_{\rho_{g,i}}/\Omega_{\rho_{s,i}}$, $x_i(t_{k+1}) \in \Omega_{\rho_{h,i}} \subset \Omega_{\rho_1}$.

To ensure that the estimate for $t \in [t_k, t_{k+1})$ is within $\Omega_{\rho_i} \subset \Omega_{\rho_1}$, Proposition 2 gives the following inequality:

$$V_i\left(z_{1,i}(t_{k+1})\right) \le V_i(x_i(t_{k+1})) \\ + f_{V,i}\left(|x_i(t_{k+1}) - z_{1,i}(t_{k+1})|\right) \le V_i(x_i(t_{k+1})) + f_{V,i}\left(\epsilon^*_{M,i}\right) \tag{58}$$

When $x_i(t_{k+1}) \in \Omega_{\rho_{h,i}}$ as was just demonstrated for the case that no attacks occur, this gives that $V_i(z_{1,i}(t_{k+1})) \le \rho_i$ if **Eq. 46** holds. If instead $x_i(t_k) \in \Omega_{\rho'_{s,i}}$, **Eq. 47** ensures that $V_i(x_i(t_{k+1})) \in \Omega_{\rho_{\min,i}} \subset \Omega_{\rho_{h,i}}$ and therefore we conclude by the same logic as above that $V_i(z_{1,i}(t_{k+1})) \le \rho_i$ if **Eq. 46** holds.

To see that the implementation strategy with updates of i and the LEMPC at every sampling time maintains $x_i(t_k) \in \Omega_{\rho_1}$ and $z_{1,i}(t_k) \in \Omega_{\rho_1}$ for all time, we note that the proof above shows that if $x_i(t_{z1}) \in \Omega_{\rho_{h,1}}$, then $z_{1,i}(t_{k+1}) \in \Omega_{\rho_1}$ and $x_i(t_{k+1}) \in \Omega_{\rho_{h,i}} \subset \Omega_{\rho_1}$. At t_{k+1}, under the assumption of the theorem that it is again possible to find all regions for LEMPC design according to Step 4 of the implementation strategy, Ω_{ρ_i} and its subsets will be selected so that the same proof as above holds throughout the subsequent sampling period and $z_{1,i}(t_{k+2}) \in \Omega_{\rho_1}$ and $x_i(t_{k+2}) \in \Omega_{\rho_{h,i}} \subset \Omega_{\rho_1}$.

This indicates that $z_{1,i}$ would be within Ω_{ρ_1} at all sampling times before an attack, and that $x_i(t) \in \Omega_{\rho_1}$ as well. To ensure $(V_i(z_{1,i}(t_{k+1}))-V_i(z_{1,i}(t_k))) < 0$ so that flagging an attack in Step 6 of the implementation strategy of **Section 7.1** would not cause attacks to be detected when none are occurring, the requirement of **Eq. 49** with the input computed by the *i*-LEMPC should be satisfied, according to the logic of **Section 4.1** of this manuscript.

Part 4. Finally, we consider the case that at some $t_k \geq t_q$, the process is under either an undetected false sensor measurement cyberattack (Case 1), actuator cyberattack (Case 2) or both (Case 3).

Part 4—Case 1. If the control system is under only a sensor attack, but it is not detected, $|z_{1,i}(t_k) - x_i(t_k)| \leq \epsilon^*_{M,i}$. Thus, from Part 3 above, the closed-loop state and state estimate are guaranteed to be inside $\Omega_{\rho_i} \subset \Omega_{\rho_1}$ by the implementation strategy of **Section 7.1**.

Part 4—Case 2. If the control system is under only an actuator attack, but it is not detected, then an input that is not that computed by the *i*-LEMPC is being applied to the process over a sampling period. The actuator attack will be detected if several conditions that are evaluated at the end of a sampling period (at t_{k+1}) occur [e.g., $V_j(z_{1,i}(t_k)) < V_j(z_{1,i}(t_{k+1}))$, $V_j(z_{1,i}(t_{k+1})) > \rho_i$, $|z_{j,i}(t_{k+1}) - z_{p,i}(t_{k+1})| > \epsilon_{\max}$, $j = 1, \ldots, M$, $p = 1, \ldots, M$, or $V_j(z_{1,i}(t_{k+1})) \notin \Omega_{\rho_1}$]. However, if an actuator attack occurs at t_k, this means that its effects will not be observed for flagging an attack until t_{k+1}, leaving the possibility that the closed-loop state could exit a desired operating region before the sampling period is over. To prevent this, we define a worst-case scenario in **Eqs 50**, **51**, where it may be possible that the state estimate is within Ω_{ρ_1} at a sampling time but that the actual state is outside of it (within $\Omega_{\rho_{samp3}}$) and an attack is not flagged since $V_j(z_{1,i}(t_k)) \in \Omega_{\rho_1}$ (i.e., at least one of the detection conditions is not violated, leaving a possibility of non-detection depending on the state of the other detection conditions). In such a case, under a rogue actuator output, the closed-loop state either remains in $\Omega_{\rho_{samp3}}$, where the estimate may not be outside Ω_{ρ_1} for detecting the attack based on whether $z_{1,i}(t_k) \in \Omega_{\rho_1}$ or not, or it is within $\Omega_{\rho_{samp4}} \subset \Omega_{\rho_{safe}}$, but in a part of it where the attack can be flagged at t_{k+1}. Then, the attack is flagged while the closed-loop state is still in $\Omega_{\rho_{safe}}$. In contrast, if the state estimate was in Ω_{ρ_1}, then in the following sampling period, the closed-loop state either enters $\Omega_{\rho_{samp4}}/\Omega_{\rho_{samp3}}$ and is flagged, or it remains in $\Omega_{\rho_{samp3}}$ and this process continues into subsequent sampling periods. The attack would be flagged before the closed-loop state leaves $\Omega_{\rho_{safe}}$ because **Eqs 50**, **51** show that the state cannot go farther from the origin than $\Omega_{\rho_{samp4}}$ in a sampling period if the attack is not detected at the beginning of the sampling period, and $\Omega_{\rho_{samp4}} \subset \Omega_{\rho_{safe}}$.

Part 4—Case 3. If the control system is under both sensor and actuator attacks, but they are not detected, the rogue actuator and sensor outputs must still maintain the state estimates in Ω_{ρ_1}. Since the state estimates must be within Ω_{ρ_1} and $|z_{1,i}(t_k) - x_i(t_k)| \leq \epsilon^*_{M,i}$ must be satisfied (as a sensor attack is not detected) with at least one estimate not being affected by an attack, the reasoning in Part 4—Case 2 can be used to conclude that the combined attacks cannot cause the closed-loop state or state estimate to exit $\Omega_{\rho_{safe}}$ without the attack being detected.

Above, it is demonstrated that whether attacks are occurring or not, the closed-loop state and state estimate cannot leave $\Omega_{\rho_{safe}}$ without an attack being detected in any sampling period. This indicates that the implementation strategy in **Section 7.1** maintains the closed-loop state within a safe operating region at all times before an attack is detected, even if undetected sensor and actuator attacks occur during that time period.

Remark 9. The proof for Part 4—Case 2 described above gives an indication of how the proof of closed-loop stability for actuator-only attacks on an LEMPC of the 1-A form would be carried out, but (noisy) state measurements then might be used in place of state estimates.

Remark 10. Several regions have been defined for the proposed detection strategy. $\Omega_{\rho_i} \subset \Omega_{\rho_{safe}}$, $i = 1, 2, \ldots$, has been defined as an invariant set in which the closed-loop state is maintained. We define the region $\Omega_{\rho_{h,i}}$ such that if the state measurement is within $\Omega_{\rho_{g,i}}$ at t_k, the actual state is within $\Omega_{\rho_{h,i}} \subset \Omega_{\rho_i}$ (**Eq. 43**). We also define the region $\Omega_{\rho_{s,i}}$ such that if the state measurement is within $\Omega_{\rho_i}/\Omega_{\rho_{s,i}}$ at t_k, the actual state is not within $\Omega_{\rho'_{s,i}}$ (**Eq. 48**). In addition, $\Omega_{\rho_{\min,i}}$ is characterized as a region where if $x_i(t_k) \in \Omega_{\rho'_{s,i}}$, the actual state is within $\Omega_{\rho_{\min,i}}$ (**Eq. 47**). The definition of $\Omega_{\rho_{h,i}}$ ensures that the state estimate at t_{k+1} is in Ω_{ρ_i} when there is no attack, if $x_i(t_k) \in \Omega_{\rho_{h,i}}$.

7.2 Simultaneous Sensor and Actuator Attack-Handling *via* Detection Strategies 3-S and 2-A: Formulation and Implementation

Following the idea of pairing single detection strategies above, another integrated framework, named Detection Strategy 2/3, can be developed that uses redundant state estimates to check for sensor attacks (again assuming that at least one of the estimates is not impacted by any attack) and relies on the difference between a state prediction based on the last available state estimate (obtained using an expected control action computed by either a fully redundant controller or an approximation of the controller output for a given state estimate) and a state estimate being less than a bound. The premise of checking the difference between the state estimate and the state prediction is that the state prediction should not be able to deviate too much from a (converged) state estimate (i.e., it approximates the actual process state to within a bound as in Assumption 2 after a sufficient period of time has passed since initialization of the state estimates) if there are no sensor or actuator attacks, and that therefore, seeing the estimate and prediction deviate by more than an expected amount is indicative of an attack.

If the actual state is inside a subset $\Omega_{\rho_{\max}}$ of the stability region Ω_ρ, then under sufficient conditions (which will be clarified in the next section), both the closed-loop state and state estimate are maintained in a safe operating region $\Omega_{\rho_{safe}}$ for all time for the process without attacks or with undetected attacks. The notation to be used for the LEMPC for Detection Strategy 2/3 follows that of **Eq. 10** with **Eq. 10c** replaced by $\tilde{x}(t_k) = z_1(t_k)$ (we will subsequently refer to this LEMPC as the output feedback LEMPC of **Eq. 10**. In this control formulation, the output feedback LEMPC design of **Eq. 10** receives a state estimate z_j ($j = 1, \ldots, M$) from one of the redundant state estimators (the estimator used to provide state estimates to the proposed LEMPC that controls the process will be denoted as the $j = 1$ estimator) at t_k.

To present the implementation strategy and subsequent proof in the next section that Detection Strategy 2/3 can be made cyberattack resilient in the sense that it can guarantee safety whenever no sensor or actuator attacks are flagged by this combined detection framework, it is necessary to determine the detection threshold for the difference between the state estimate and state prediction. Unlike in the case where a bound on the difference between state predictions and state measurements was derived for Detection Strategy 2-S for sensor attacks, we here need to set up mechanisms for detecting whether an actuator and/or sensor attack occurs. While state estimates are available to aid in detecting sensor attacks, a part of the mechanism for detecting whether actuator attacks occur is the use of a fully redundant controller (for which the input computed by the output feedback LEMPC of **Eq. 10** is equivalent to the input computed by the redundant controller used in cross-checking the controller outputs) or the fast approximation of the control outputs (for which the input computed by the LEMPC would differ, within a bound, from the input computed by the algorithm used in cross-checking the controller outputs) for a given state measurement. The definition below defines the notation that will be used in this section to represent the actual state trajectory under the control input computed by the LEMPC and the state prediction obtained from the nominal ($w \equiv 0$) process model of **Eq. 1** under the potentially approximate input used for cross-checking the control outputs.

Definition 1. Consider the state trajectories for the actual process and for the predicted state from $t \in [t_0, t_1)$, which are the solutions of the systems:

$$\dot{x}_a = f(x_a(t), \bar{u}(t), w(t)) \tag{59a}$$

$$\dot{x}_b = f(x_b(t), \hat{u}(t), 0) \tag{59b}$$

where $|x_a(t_0) - z_1(t_0)| \le \gamma$. x_a is the state trajectory for the actual process, where $\bar{u}$ is the optimal input for $t \in [t_0, t_1)$ computed from the output feedback LEMPC of **Eq. 10** based on the estimate $z_1(t_0)$, where $z_1(t_0)$ is an estimate of the actual state $x_a(t_0)$ at t_0. $\hat{u}$ is a (potentially) different input that is applied to the process that results in the trajectory x_b corresponding to the predicted value of the closed-loop state when $\hat{u}$ is computed by the method for cross-checking the controller inputs. For any method used for cross-checking the controller inputs computed, the following bound is assumed to be known to hold:

$$|\bar{u}(t) - \hat{u}(t)| \le \epsilon_u \tag{60}$$

where ϵ_u is the maximum deviation in the inputs computed for a given state estimate between the output feedback LEMPC of **Eq. 10** and the method for cross-checking the controller inputs (if a fully redundant controller is utilized, $\epsilon_u = 0$).

The following proposition bounds the difference between x_a and x_b in Definition 1.

Proposition 4. Consider the systems in Definition 1 operated under the output feedback LEMPC of **Eq. 10** and designed based on a controller $h(\cdot)$, which satisfies **Eqs 2**, **3**. Then, the following bound holds:

$$|x_a(t) - x_b(t)| \le f_u(\gamma, t) \tag{61}$$

and initial states $|x_a(t_0) - x_b(t_0)| \le \gamma$, where $x_b(t_0) = z_1(t_0)$ and $t_0 = 0$:

$$f_u(s, \tau) := s e^{L_x t} + \left(e^{L_x t} - 1\right)\left(\frac{L_u \epsilon_u + L_w \theta}{L_x}\right) \tag{62}$$

Proof 2. Integrating **Eqs 59a**, **59b** from t_0 to t, subtracting the second equation from the first, and taking the norm of both sides gives

$$|x_a(t) - x_b(t)| \le |x_a(t_0) - z_1(t_0)| + \int_0^t |f(x_a(s), \bar{u}(0), w(s)) - f(x_b(s), \hat{u}(0), 0)| \, ds \tag{63a}$$

$$\le \gamma + \int_0^t \left[|f(x_a(s), \bar{u}(0), w(s)) - f(x_b(s), \bar{u}(0), 0)| + |f(x_b(s), \bar{u}(0), 0) - f(x_b(s), \hat{u}(0), 0)|\right] ds \tag{63b}$$

for $t \in [0, t_1)$. Using **Eqs 4a**, **4c** and the bound on w, the following bound is achieved:

$$|x_a(t) - x_b(t)| \le \gamma + \int_0^t \left[L_u|\bar{u}(0) - \hat{u}(0)| + L_x|x_a(s) - x_b(s)| + L_w|w(s)|\right] ds \tag{64a}$$

$$\le \gamma + L_u|\bar{u}(0) - \hat{u}(0)|(t - 0) + L_x \int_0^t |x_a(s) - x_b(s)| + L_w \int_0^t \theta \, ds \tag{64b}$$

$$\le \gamma + (L_u \epsilon_u + L_w \theta)t + L_x \int_0^t |x_a(s) - x_b(s)| \, ds \tag{64c}$$

for $t \in [0, t_1)$, where the last inequality follows from **Eq. 60**. Finally, using the Gronwall–Bellman inequality Khalil (2002), it is obtained that

$$|x_a(t) - x_b(t)| \le \gamma e^{L_x t} + \left(e^{L_x t} - 1\right)\left(\frac{L_u \epsilon_u + L_w \theta}{L_x}\right) \tag{65}$$

Proposition 4 can be used to develop an upper bound on the maximum possible error that would be expected to be seen between a state prediction and a state estimate at a sampling time if no attacks occur. This bound is developed in the following proposition.

Proposition 5. Consider x_a and x_b defined as in Definition 1. If $|z_j(t_k) - z_p(t_k)| < \epsilon_{\max}$ and $|z_j(t_{k+1}) - z_p(t_{k+1})| < \epsilon_{\max}$, j = 1, . . . , M, p = 1, . . . , M, and **Eq. 60** holds in the absence of an attack, then the worst-case error between the state estimate $z_1(t_{k+1})$ and the state prediction $\tilde{x}_b(t_{k+1}|t_k)$ of the state at time t_{k+1} from an estimate obtained at time t_k in the absence of an attack on the actuators or sensors is given by

$$|z_1(t_{k+1}) - \tilde{x}_b(t_{k+1}|t_k)| \leq \epsilon_M^* + f_u(\epsilon_M^*, \Delta) \tag{66}$$

Proof 3. Using Proposition 3 and Proposition 4 along with **Eq. 32**, we obtain

$$\begin{aligned}|z_1(t_{k+1}) - \tilde{x}_b(t_{k+1}|t_k)| &\leq |z_1(t_{k+1}) - x_a(t_{k+1})| + |x_a(t_{k+1}) - \tilde{x}_b(t_{k+1}|t_k)| \\ &\leq \epsilon_M^* + f_u(|x_a(t_k) - z_1(t_k)|, \Delta) \leq \epsilon_M^* + f_u(\epsilon_M^*, \Delta)\end{aligned} \tag{67}$$

Proposition 5 demonstrates that if an upper bound $\nu_u \geq \epsilon_M^* + f_u(\epsilon_M^*, \Delta)$ is placed on $|z_1(t_{k+1}) - \tilde{x}_b(t_{k+1}|t_k)|$, then a cyberattack could be flagged if $|z_1(t_{k+1}) - \tilde{x}_b(t_{k+1}|t_k)| > \nu_u$ without creating false alarms, as $|z_1(t_{k+1}) - \tilde{x}_b(t_{k+1}|t_k)|$ should never become greater than ν_u if no attack is occurring according to the proof of the proposition.

We now describe the implementation strategy of Detection Strategy 2/3, which assumes that the process has already been run successfully in the absence of attacks under the output feedback LEMPC of **Eq. 10** for some time (t_q) such that $|z_j(t) - x(t)| \leq \epsilon_{mj}^*$ for all $j = 1, \ldots, M$.

1) At sampling time t_k, when the output feedback LEMPC of **Eq. 10** is used to control the process of **Eq. 1**, if $|z_j(t_k) - z_p(t_k)| > \epsilon_{\max}$ or $|z_j(t_{k-1}) - z_p(t_{k-1})| > \epsilon_{\max}$, $j = 1, \ldots, M$, $p = 1, \ldots, M$, or $\tilde{x}(t_k) = z_1(t_k) \notin \Omega_\rho$ (where z_1 is the state estimate used in the proposed LEMPC design that controls the process), detect that a cyberattack is occurring and go to Step 2. If no attack is flagged, check whether $|\tilde{x}(t_k|t_{k-1}) - z_1(t_k)| > \nu_u$ (where $\nu_u \geq \epsilon_M^* + f_u(\epsilon_M^*, \Delta)$). If yes, flag that a cyberattack is happening and go to Step 2. Else, go to Step 3.
2) Mitigating actions may be applied (e.g., a backup policy such as the use of redundant controller or an emergency shutdown mode).
3) Control the process using the output feedback LEMPC of **Eq. 10**. Go to Step 4.
4) ($t_k \leftarrow t_{k+1}$). Go to Step 1.

7.2.1 Simultaneous Sensor and Actuator Attack-Handling *via* Detection Strategies 3-S and 2-A: Stability and Feasibility Analysis

In this section, we prove recursive feasibility and stability of the process of **Eq. 1** under the proposed output feedback LEMPC of **Eq. 10** whenever no sensor or actuator attacks are detected according to the implementation strategy in **Section 7.2** in the presence of bounded plant/model mismatch, controller cross-check error, and measurement noise. The following theorem characterizes the safety guarantees of the process of **Eq. 1** for all time under the implementation strategy of **Section 7.2** when sensor and actuator cyberattacks are not detected. As for Detection Strategy 1/3, because the actuator cyberattacks would not be detected according to the implementation strategy in **Section 7.2** until a sampling period after they had occurred (since they are being detected by their action on the state estimates, which would not be obvious until they have had a chance to impact the closed-loop state), it is necessary to define supersets $\Omega_{\rho_{samp3}}$ and $\Omega_{\rho_{samp4}}$ of Ω_ρ, but which are contained in $\Omega_{\rho_{safe}}$, to set the size of Ω_ρ with respect to $\Omega_{\rho_{safe}}$ to ensure that Ω_ρ is defined in a sufficiently conservative fashion such that even if the closed-loop state is driven out of Ω_ρ, the closed-loop state will still always be in $\Omega_{\rho_{safe}}$ and the state estimate will go out of Ω_ρ before the actual closed-loop state leaves $\Omega_{\rho_{safe}}$.

Theorem 5. Consider the closed-loop system of **Eq. 1** under the implementation strategy of **Section 7.2**, in which no sensor or actuator cyberattack is detected using the proposed output feedback LEMPC of **Eq. 10** based on an observer and controller pair satisfying Assumption 1 and Assumption 2 and formulated with respect to the i = 1 measurement vector and a controller $h(\cdot)$ that meets **Eqs 2**, **3**. Let the conditions of Proposition 3 and Proposition 4 hold, and $\theta_w \leq \theta_w^*$, $\theta_{v,i} \leq \theta_{v,i}^*$, $\epsilon_i \in (\epsilon_{Li}^*, \epsilon_{Ui}^*)$, and $|z_i(t_0) - x(t_0)| \leq e_{m0i}$, for $i = 1, \ldots, M$. Also, let $\epsilon_{W,1} > 0$, $\Delta > 0$, $\Omega_\rho \subset X$, and $\rho_{safe} > \rho_{samp4} > \rho_{samp3} > \rho > \rho_{\max} > \rho_{1,1} > \rho'_{e,1} > \rho_{\min,1} > \rho_{s,1} > 0$, satisfy:

$$\rho'_{e,1} \leq \rho_{\max} - \max\left\{f_V(f_W(\epsilon_M^*, \Delta)), M_f \max\{t_{z1}, \Delta\}\alpha_4(\alpha_1^{-1}(\rho_{\max}))\right\} \tag{68}$$

$$\rho'_{e,1} \leq \rho - f_V(f_W(\epsilon_M^*, \Delta)) - f_V(\epsilon_M^*) \tag{69}$$

$$-\alpha_3\left(\alpha_2^{-1}(\rho_{s,1})\right) + L'_x\left(M_f\Delta + \epsilon_M^*\right) + L'_w\theta_w \leq -\epsilon_{W,1}/\Delta \tag{70}$$

$$\rho_{\min,1} = \max\left\{V(x(t+\Delta)) | V(x(t)) \leq \rho_{s,1}\right\} \tag{71}$$

$$\rho_{\min,1} + f_V(f_W(\epsilon_M^*, \Delta)) \leq \rho \tag{72}$$

$$\rho_{\max} + f_V(\epsilon_M^*) \leq \rho \tag{73}$$

$$\rho_{samp3} = \max\left\{V(x(t_k)) : z_1(t_k) \in \Omega_\rho, \ |z_1(t_k) - x(t_k)| \leq \epsilon_M^*\right\} \tag{74}$$

$$\rho_{samp4} = \max\left\{V(x(t)) : V(x(t_k)) \leq \rho_{samp3}, u(t_k) \in U, t \in [t_k, t_{k+1})\right\} \tag{75}$$

where t_{z1} is the first sampling time after t_{b1}, and f_v, f_w, and f_u are defined as in Proposition 1, Proposition 2 (with the subscripts dropped), and Proposition 4. Then, if $x(t_0) \in \Omega_{\rho_{e,1}}$, $x(t) \in \Omega_{\rho_{\max}}$ for all $t \geq 0$ and $z_1(t_h) \in \Omega_\rho$ for $t_h \geq \max\{\Delta, t_{z1}\}$ until a cyberattack is detected according to the implementation strategy in **Section 7.2**, if the attack occurs after t_q.

Proof 4. The output feedback LEMPC of **Eq. 10** has the same form as in Oyama and Durand (2020). Therefore, in the absence of attacks or in the presence of sensor attacks only, we obtain the same results as in Oyama and Durand (2020). Specifically, feasibility follows when $z_1(t_k) \in \Omega_\rho$ as proven in Oyama and Durand (2020). Since $z_1(t_k) \notin \Omega_\rho$ flags an attack according to the implementation strategy of **Section 7.2**, there will not be a time before an attack is detected that $z_1(t_k) \notin \Omega_\rho$ before an attack, so that the problem would not be infeasible before an attack. Also as demonstrated in Oyama and Durand (2020), the closed-loop state trajectory is contained in $\Omega_{\rho_{\max}}$ for $t \in [t_0, \max\{\Delta, t_{z1}\})$, and before an attack occurs when $t \geq \max\{\Delta, t_{z1}\}$, $x(t)$ is bounded within $\Omega_{\rho_{\max}}$ and $z_1(t)$ is bounded within Ω_ρ. Furthermore, it follows from Proposition 3 and Proposition 5 that the implementation strategy of **Section 7.2** will not detect measurement noise, controller cross-check error, or bounded plant/model mismatch as attacks, such that there will be no false detections. It remains to demonstrate that if there is an attack at t_k but it is not detected using the proposed methodology (i.e., $| z_i(t_k) - z_j(t_k)| \leq \epsilon_{\max}$, $|z_i(t_{k-1}) - z_j(t_{k-1})| \leq \epsilon_{\max}$, for all $i = 1, \ldots, M$, $j = 1, \ldots, M$, $\tilde{x}(t_k) = z_1(t_k) \in \Omega_\rho$, and $|\tilde{x}(t_k|t_{k-1}) - z_1(t_k)| \leq \nu_u$), then $z_1(t_{k+1})$ and $x(t)$, $t \in [t_k, t_{k+1})$, are bounded in $\Omega_{\rho_{safe}}$.

If the control system is under only a sensor attack, but it is not detected, then under the conditions of Theorem 5, the closed-loop state remains inside $\Omega_{\rho_{\max}} \subset \Omega_{\rho_{safe}}$ and the state estimate remains within $\Omega_{\rho_{safe}}$ under the implementation strategy of **Section 7.2**, following Oyama and Durand (2020).

If the control system is under only an actuator attack, then *via* the same steps as in the proof of Theorem 4 for Case 4—Part 2 with Ω_{ρ_1} replaced by Ω_ρ, the attack will be detected before it drives the closed-loop state out of $\Omega_{\rho_{safe}}$. The same proof demonstrates that when simultaneous sensor and actuator attacks occur, the closed-loop state will not be driven out of $\Omega_{\rho_{safe}}$ before an attack is detected. Applying these proofs recursively indicates that under this implementation strategy, an attack is detected before the closed-loop state leaves $\Omega_{\rho_{safe}}$.

Remark 11. The proof for actuator-only attacks for Theorem 5 described above gives an indication of how the proof of closed-loop stability for actuator-only attacks on an LEMPC of the 2-A form would be carried out, but state measurements might then be used in place of state estimates, with the bound developed on the difference between the state estimate and state prediction updated to be between the measurement and prediction.

8 CYBERATTACK DISCOVERABILITY FOR NONLINEAR SYSTEMS

The above sections reviewed a variety of cyberattack-handling mechanisms that rely on specific detection strategies designed in tandem with the controllers. None of those strategies, in the manner discussed, detects every attack, but some ensure that safety is maintained when the attacks are not detected. This raises the question of when detection mechanisms can detect attacks and when they cannot. This section is devoted to a discussion of these points. In Oyama et al. (2021), we first presented the notions of cyberattack discoverability for nonlinear systems in a discussionary sense (i.e., a stealthy attack is fundamentally "dynamics-based" or a "process-aware policy" and could fly under the radar of any reasonable detection method; on the other hand, a "non-stealthy" attack can be viewed as the one in which the attack policy is not within the bounds of a detection threshold and could promptly be flagged as a cyberattack using a reasonable detection method). In this section, we present the mathematical characterizations of nonlinear systems cyberattack discoverability that allow us to cast the various attack detection and handling strategies explored in this work in a unified framework and to more deeply understand the principles by which they succeed or do not succeed in attack detection.

We begin by developing a nonlinear systems definition of cyberattack discoverability as follows:

Definition 2. (Cyberattack Discoverability): Consider the state trajectories from $t \in [t_0, t_1)$ that are the solutions of the systems:

$$\dot{x}_a(t) = f(x_a(t), u_a(x_0 + \nu_a), w_a(t)) \tag{76}$$

$$\dot{x}_b(t) = f(x_b(t), u_b(x_0 + \nu_b), w_b(t)) \tag{77}$$

where $u_a(x_0 + \nu_a)$ and $u_b(x_0 + \nu_b)$ are the inputs to the process for $t \in [t_0, t_1)$ computed from a controller when the controller receives a measurement $\tilde{x}_a(t_0) = x_0 + \nu_a$ (with $|\nu_a| \leq \theta_{\nu_a}$) or $\tilde{x}_b(t_0) = x_0 + \nu_b$ (with $|\nu_b| \leq \theta_{\nu_b}$), respectively. If a reasonable detection method would be able to distinguish between the x_a and x_b trajectories, then the system is said to be cyberattack discoverable. Otherwise, it is said to be cyberattack undiscoverable.

This definition of cyberattack discoverability is related to whether multiple valid measurements or multiple valid inputs could be measured or could be possible from a given state at a certain time, obscuring whether what is presented to the detection algorithm is correct. Cyberattacks can involve deliberate changes of the information that might make them observable. Detecting a cyberattack purely from process physics data may be challenging because it requires developing the "expectations" of what the process data should be, which should be derived either from experience or a model. If the data from which predictions are made or conclusions are drawn are falsified, it may be difficult to determine the appropriate expectation.

We now present a number of comments on the methods discussed in this work and how these methods can be understood in light of a broader discoverability context:

- If there are sensor attacks only, the functions u_a and u_b in Definition 2 may be the same, with the different arguments $x_0 + \nu_a$ and $x_0 + \nu_b$. If an actuator only is attacked, $x_0 + \nu_a$ and $x_0 + \nu_b$ can be the same.
- The detection strategies presented in this work have implicitly relied on Definition 2. They have attempted, when an attack would cause a safety issue, to force that attack to be discoverable, by making, for example, the state measurement under an expected control action $u_a(x_0 + \nu_a)$ different from the state measurement under a rogue policy $u_b(x_0 + \nu_b)$. We have seen methods fail to detect attacks when they cannot force this difference to appear. This fundamental perspective has the benefit of allowing us to better understand where the benefits and limitations of each of the methods arise from, which can guide future work by suggesting what aspects of strategies that fail would need to change to make them viable.
- The definition presented in this section helps to clarify the question of what the fundamental nature of a cyberattack is, in particular a stealthy attack, that may distinguish it from disturbances. Specifically, consider a robust controller designed to ensure that any process disturbance within the bounds of what is allowed for the control system should maintain the closed-loop state inside a safe region of operation for all time if no attack is occurring. In other words, the plant–model mismatch is accounted for during the control design stage and does not cause the feedback of the state to be lost. However, a stealthy attack is essentially a process-aware policy or an intelligent adversary that can

modify the sensor measurements and/or actuator outputs through attack policies with a specific goal of making it impossible to distinguish between the actual and falsified data. The result of this is that stealthy attacks could fly under the radar of any reasonable detection mechanism and thus the control actions applied to the process may not be stabilizing. We have previously examined an extreme case of an undiscoverable attack in Oyama et al. (2021), where the attack was performed on the state measurements of a continuous stirred tank reactor by generating measurements that followed the state trajectory that would be taken under a different realization of the process disturbances and measurement noise and providing these to the controller. This would make the stealthy sensor attack, at every sampling time, appear valid to a detection strategy that is not generating false alarms.

- If a system is continuously monitored before and after an attack and the pair $\{w_b, v_b\}$ does not follow the same disturbance and noise distribution as the pair $\{w_a, v_a\}$ in Definition 2, a cyberattack could conceivably be flagged by a detection method that is able to discern that.
- We note that although Definition 2 implies that if the attacker knows the process model and disturbance and noise distributions, they could implement an attack policy such that x_a and x_b trajectories cannot be distinguished (in the sense that one cannot be flagged as abnormal); cyberattack undiscoverability does not necessarily imply the loss of closed-loop stability. Specifically, if the closed-loop state trajectory (x_a) and the false closed-loop state trajectory (x_b) are "close enough" such that a Lyapunov function decreases along the closed-loop state trajectory in both cases under the inputs computed for both, then the closed-loop state may still be maintained within a desired operating region under the attack. This is implied by the fact that conservatively designed controllers can handle sufficiently small measurement noise (as, for example, in Detection Strategy 3-S described above). As a further example, consider that an attacker seeks to develop a falsified state measurement trajectory using disturbances w_2 that are in the same distribution as those (w_1) impacting the actual process for a closed-loop system under an explicit control policy $h(x)$ with full-state feedback:

$$\dot{x}_1(t) = f(x_1(t), h(x_2(t)), w_1(t)) \tag{78}$$

$$\dot{x}_2(t) = f(x_2(t), h(x_2(t)), w_2(t)) \tag{79}$$

Depending on the trajectories of w_1 and w_2 (i.e., how the attacker's simulated noise/disturbance profile deviates from that which is experienced by the true process over time), the closed-loop system of **Eqs 78**, **79** may maintain x_1 in a bounded operating region (i.e., it may be stabilizing for the actual process system) or it may not. A nonlinear systems analysis [*via*, for example, the Lyapunov stability theory for the different potential functions of $w_1(t)$ and $w_2(t)$] could be used to evaluate what types of disturbance/noise realizations and corresponding falsified conditions would enable a "dynamics-based" attack with this structure to be destabilizing. This is the same conclusion as was drawn in cases where one of the detection strategies described above was not effective at detecting an attack; many of the undetected attacks described did not prevent safety issues, which was the premise of the simultaneous actuator and sensor detection policies.

- Definition 2 assumes that no change in the process dynamics occurs. If the process dynamics change over time, the state trajectories, which are the solutions of the system indicated in the cyberattack discoverability definition, may significantly differ from the state trajectories prior to this change. If the detection scheme would then be set up to compare expectations under the old and new process models, the change in process dynamics may be erroneously flagged as a cyberattack. In Rangan et al. (2021), for example, we provide a two-fold control/detection mechanism to prevent false attack detection when the variations in the process dynamics are considered.
- Though methods for making cyberattacks discoverable might benefit from the knowledge of the distribution of the noise and disturbances (to better distinguish **Eqs 76**, **77**), the various detection strategies developed in this work make no consideration for statistics; they look only at the bounds on disturbances and sensor measurements. The only requirements made on the attacks are that the sensor measurement cyberattacks keep the state measurements in the regions that do not flag the attacks (e.g., subsets of the stability region), and that the inputs remain in the input bounds (which must be true physically). Strategies such as those described in **Sections 7.1, 7.2** were demonstrated to avoid false-positive detections of attacks by using these bounds instead of distributional information for the noise and disturbances.

9 PROBING THE PRACTICALITY OF LEMPC-BASED CYBERATTACK-RESILIENT CONTROL DESIGN

The results above suggest that if controllers can be designed to satisfy the theoretical requirements discussed in the prior sections, there would be benefits to using them from a cybersecurity perspective. However, an important question that arises from these studies is how easy it might be to design controllers satisfying the theoretical requirements (and if it would be practical at all) and what the answer to this question suggests about how the future work in cyberattack-resilient LEMPC should continue. In our prior work (Oyama et al. 2021), a number of simulations of a sensor measurement cyberattack-handling LEMPC that can also account for the changes in the process dynamics were performed. The results indicated that checking that the parameters of the control law and detection strategy (such as thresholds used in the detection policy or ρ_e) prevent cyberattacks from being successful can be challenging if it is performed using only a limited number of simulations. This suggests that either a significant number of simulations may be needed to design cyberattack-resilient LEMPCs (which would be expected to be a challenging way to design these controllers due to the interactions between the various parameters and could also still potentially leave system vulnerabilities if the simulations are not able to fully cover every possible issue), or a method for obtaining the parameters of the LEMPCs that meet the theory would be needed.

In this section, we seek to provide some initial insights into obtaining parameters for an LEMPC that meet the theory. To make progress on this, we remove some of the complexity of the problem by focusing on how to obtain the theoretical parameters not of the more specialized LEMPCs for cyberattack-resilient control discussed in this work, but instead for the original LEMPC developed in Heidarinejad et al. (2012a). This discussion is used to motivate future work in seeking to extend the initial results presented here on obtaining LEMPC parameters to more comprehensive methods for obtaining these parameters that could then be scaled to the cyberattack-resilient forms of LEMPC to eliminate the vulnerabilities.

9.1 LEMPC: Meeting Theoretical Requirements in Control Design

Before moving to a study working toward obtaining LEMPC parameters for a CSTR example, we first discuss a number of preliminaries regarding this topic. First, since this section will focus on the standard LEMPC of **Eqs 8**, **9**, instead of its cyberattack-resilient form, we consider Proposition 2 (where in the remainder of this section; we will neglect the subscript i for the simplicity of notation) and the following proposition and theorem.

Proposition 6. Mhaskar et al. (2012), Heidarinejad et al. (2012a) Consider the following two systems:

$$\dot{x}_a = f(x_a(t), u(t), w(t)) \tag{80a}$$

$$\dot{x}_b = f(x_b(t), u(t), 0) \tag{80b}$$

with initial states of $x_a(t_0) \in \Omega_\rho$ and $x_b(t_0) \in \Omega_\rho$. There exists a class $\mathcal{K}$ function $f_W(\cdot)$ that satisfies the following equations $\forall\ x_a, x_b \in \Omega_\rho$ and $\forall\ w \in W$:

$$|x_a(t) - x_b(t)| \le \bar{f}_W(t - t_0) \tag{81a}$$

$$\text{where } \bar{f}_W(\tau) := \frac{L_w \theta_w}{L_x}\left(e^{L_x \tau} - 1\right) \tag{81b}$$

Theorem 6. Heidarinejad et al. (2012a) Consider the system of **Eq. 1** in closed loop under the LEMPC design of **Eqs 8**, **9** based on a controller $h(x)$ that satisfies the conditions of **Eq. 2**. Let $\epsilon_w > 0$, $\Delta > 0$, and $\rho > \rho_e > \rho_{\min} > \rho_s > 0$ satisfy

$$\rho_e' \le \rho - f_V\left(\bar{f}_W(\Delta)\right) \tag{82}$$

and

$$-\alpha_3\left(\alpha_2^{-1}(\rho_s)\right) + L_x' M_f \Delta + L_w' \theta_w \le -\epsilon_w \big/ \Delta \tag{83}$$

where

$$f_V(s) = \alpha_4\left(\alpha_1^{-1}(\rho)\right)s + M_v s^2 \tag{84}$$

for M_v as a positive constant. If $x(t_0) \in \Omega_\rho$ and $N \ge 1$ where

$$\rho_{\min} = \max\{V(x(t)) : V(x(t_k)) \le \rho_s,\ t \in [t_k, t_{k+1}),\ u(t_k) \in U\} \tag{85}$$

then the state $x(t)$ of the closed-loop system is always bounded in Ω_ρ and is ultimately bounded in $\Omega_{\rho_{\min}}$.

The conditions of Theorem 6 involve many functions and parameters that must relate to one another in a specific way. Finding all of these functions and parameters has the potential to be somewhat cumbersome, particularly for larger systems. For example, from **Eq. 83**, it can be seen that Δ cannot be too large (or else the left-hand side of **Eq. 83** will not be negative); however, what "too large" means is unclear. One idea for attempting to satisfy the theory is to set up mechanisms for moving the parameters in desirable directions (e.g., smaller values of Δ), hoping that will be "enough." One idea like this was explored in our prior work Durand and Messina (2020). In that work, we focused specifically on the relationship between ρ_e' and Δ. From **Eq. 82**, it can be seen that larger values of ρ_e' require smaller values of Δ; however, how large ρ_e' can be for a given value of Δ is not obvious without obtaining all controller parameters to ensure that they meet the set of all equations in Theorem 6. As the sampling period approaches 0, the value of ρ_e' might be able to be made larger while retaining stability guarantees.

In practice, the value of Δ will always be nonzero and is generally limited by the computation time of the LEMPC. However, we consider that there may be more frequent measurements from sensors than the frequency of the LEMPC computation. Therefore, in Durand and Messina (2020), we suggested attempting to utilize a desired ρ_e' in the LEMPC, and then to use sensor measurements obtained multiple times throughout each Δ and activating a back-up explicit stabilizing controller capable of driving the closed-loop state toward the origin when the closed-loop state leaves $\Omega_{\rho_e'}$. Due to the increased frequency of measurements, the amount of time that may elapse between the time the closed-loop state leaves $\Omega_{\rho_e'}$ and the next sensor measurement that detects the departure is decreased, which may allow $\Omega_{\rho_e'}$ to take a wider range of values compared to the standard LEMPC formulation. However, despite the fact that this is a possibility, this still does not rigorously address how to develop an LEMPC that meets the theoretical requirements and is therefore not a method that would be expected to translate to a cyberattack-resilient LEMPC design.

One of the first steps in designing an LEMPC design according to the theory is obtaining functions such as V and h. A variety of studies have been performed related to designing Lyapunov functions and stabilizing control laws. For example, $h(x)$ could be designed with methods such as the linear quadratic regulator (LQR) (Bemporad et al., 2002; Griffith, 2018) or Sontag's formula Lin and Sontag (1991) (the latter for input-affine process models).

Methods have been explored for constructing Lyapunov functions such as sum of squares (SOS) decomposition (Papachristodoulou and Prajna, 2002). For an LEMPC, it is not only functions such as V and h that must be found, but also other functions such as α_i, i = 1, 2, 3, 4, such that all conditions of Theorem 6 are satisfied.

However, to design a "good" LEMPC meeting Theorem 6, we would like to find parameters such as h and V that have special properties; in particular, we would like them to cause the LEMPC to: 1) have parameters such as Δ that allow it to be physically implemented on existing systems; and 2) provide significant profit (the most possible with physically-implementable versions of the parameters). In the study in the next section, we will start with an assumed h, V, and ρ, and then see what values parameters such as Δ would take, to see if they are physically realizable. This will provide insight into some potential challenges of practically designing an LEMPC where the theory is met.

Remark 12. As a comment on the last paragraph above, we remark that the requirements noted form a sort of optimization problem for h and V. To gain insight into the task, we could ask whether it would be possible to form the set of every possible h (Lipschitz continuous functions) and V (positive definite functions) and then to search within this set for h and V combinations that not only satisfy fundamental objectives of these functions (such as satisfaction of **Eqs 2a**, **2b**) but which also enable the resulting h and V to cause all other parameters of the LEMPC to satisfy the two objectives of the LEMPC in the prior paragraph. We might begin by considering suggesting forms of h and V and then finding their form *via* optimization based on techniques in Brunton et al. (2016). Specifically, Brunton et al. (2016) develops potential dynamic models by guessing the terms that may appear on the right-hand side and then attempting to use a sparse regression to locate which of those should be used to represent the process dynamics. This begs the question of whether a form for h might be guessed, and then an optimization problem solved in which the coefficients of the terms of the form of h are the decision variables and the constraints enforce $\dot{V}$ to be negative at many points in a discretization of the state-space, to determine a form of h systematically. Because this relates h and V to an optimization problem, a method like this might have flexibility to then be combined with other strategies for optimizing the h and V choice to attempt to achieve the goals in the paragraph above. However, even for this preliminary optimization problem concept that does not explicitly account for those alternative goals, without careful structuring, the resulting optimization problem is not guaranteed to be feasible. We can analyze this from a fundamental control-theoretic perspective. First, we note that for a given discretization of the state-space, there does not necessarily exist any input policy that, at all points in the state-space, can drive the closed-loop state to the origin (this only occurs within the region of attraction). Second, even if the discretization of the state-space being examined only includes the region of attraction, the input trajectory that could drive the closed-loop state to the origin from a given initial condition in that portion of the state-space does not necessarily stay within that discretized region or cause a given V to decrease (i.e., the region of attraction is independent from V). Therefore, guessing a form of h to search for a control design that might make $\dot{V}$ negative *via* optimization of its terms (with the subsequent goal of modifying the problem to account for other goals we would like to achieve with these functions) may have limitations. Even if it was possible to suggest a form of h that could approximate many functions, for each V, there is an upper bound on it where the level set is in the region of attraction (it is not possible to consider beyond that ρ). The question asked is which h and V combination with an upper bound on V below the threshold for that V gives the maximum EMPC profit and implementable parameters. This could be explored in a brute force fashion by looking at every possible value of V, for each finding the maximum value of ρ, testing it for every possible value of h, obtaining the resulting control parameters, and seeing the best profit among those with reasonable control parameters, and selecting the one with the best profit. The challenge with doing this is the need to test every point and every function (and then also there is no guarantee that practically implementable parameters will be obtained). If there is a finite set of h's, it is not guaranteed that there is one that is stabilizing in that set. The guarantee is that there is some trajectory u that is stabilizing in the region of attraction, but whether that includes the ones that are allowed once the function is parametrized is not guaranteed. This discussion indicates that considering how to obtain optimal and practical designs of LEMPCs will require many questions to be addressed beyond what is presented in the subsequent section as a preliminary step in moving toward developing LEMPCs with parameters related to the full control theory.

9.1.1 Obtaining Control-Theoretic Parameters for LEMPC Applied to a CSTR

In this section, we provide a brute force-type method for exploring the parameters of an LEMPC that might be more aligned with the theory than assumed values. The brute force-type approach does not ensure that all of the parameters meet the theory, but it provides many insights into the shortcomings of this initial approach for attempting to obtain the parameters to motivate further studies on this topic and potential challenges with the parameters that might be obtained.

We consider the nonlinear process model of **Eqs 40**, **41**. The manipulated inputs are C_{A0} (the reactant feed concentration of species A) and Q (the heat rate input), with the bounds of $0.5 \le C_{A0} \le 7.5$ kmol/m^3 and $-5.0 \times 10^5 \le Q \le 5.0 \times 10^5$ kJ/h. The values of the parameters of the CSTR model are presented in **Table 2**. An open-loop asymptotically stable steady-state occurs at C_{As} = 1.2 kmol/m^3 and T_s = 438.2 K, where the subscript s indicates the steady-state values. In the control formulation, the state and input vectors are represented using deviation variables as $x^T = [C_A - C_{As} \; T - T_s]$ and $u^T = [C_{A0} - C_{A0s} \; Q - Q_s]$, respectively.

According to Theorem 6, the first step in finding the control-theoretic parameters for LEMPC is to find a controller $h(x)$ satisfying **Eq. 2** so that Ω_ρ, V, and h in the LEMPC of **Eqs 8**, **9** can be defined. In general, it may be challenging to find the functions α_1, α_2, α_3, α_4, and $h(x)$ satisfying the requirements of **Eq. 2**. The input-affine form of **Eqs 40**, **41** allows Sontag's formula (Lin and Sontag 1991) to be used for $h(x)$ (assuming

TABLE 2 | Parameters for the CSTR model.

Parameter	Value	Unit
V	1	m^3
T_0	300	K
C_p	0.231	kJ/kg·K
k_0	8.46×10^6	m^3/h·kmol
F	5	m^3/h
ρ_L	1,000	kg/m^3
E	5×10^4	kJ/kmol
R	8.314	kJ/kmol·K
ΔH	-1.15×10^4	kJ/kmol

$\hat{h}_1 = u_1 = 0$ kmol/m^3 and that Sontag's formula is then used only for $\hat{h}_2$) with a guaranteed decrease on $V(x)$, and an attempt to use a quadratic form of V as x^TPx with a positive definite P makes it possible to find some α_1 and α_2 satisfying **Eq. 2a** (if that selection of V turns out to be successful; notably, however, the fact that these functions satisfy some of the equations does not mean that they will make it possible or straightforward to satisfy the others). The manner in which we proceed here is as follows: initially, we select a quadratic form of V with $P = [2{,}000\ -10;\ -10\ 3]$. For a symmetric P, $\lambda_{\min}(P)x^Tx \le x^TPx \le \lambda_{\max}(P)x^Tx$, where $\lambda_{\min}(P)$ and $\lambda_{\max}(P)$ represent the minimum and maximum eigenvalues of P, respectively. This indicates that for a symmetric P utilized for $V = x^TPx$, $\alpha_1(|x|)$ can be set to $\lambda_{\min}(P)|x|^2$, and $\alpha_2(|x|)$ can be set to $\lambda_{\max}(P)|x|^2$. For the given P, $\lambda_{\min}(P)$ and $\lambda_{\max}(P)$ can be found using MATLAB's eig function to be 2.95 and 2,000.05, respectively. From this, we will set $a_1 = 2.9$ and $a_2 = 2001$, where $\alpha_1(|x|) = a_1|x|^2$ and $\alpha_2(|x|) = a_2|x|^2$.

The next function that we would like to obtain is α_3. According to **Eq. 2b**, $\alpha_3(|x|)$ should be a class $\mathcal{K}$ function that provides an upper bound to $\dot{V}$ along the closed-loop state trajectories at all points in the stability region. While it would be ideal in general to find such a function analytically, we perform an approximate check numerically here using $\hat{h}_1(x) = 0$ kmol/m^3 and $\hat{h}_2(x)$ given by Sontag's formula (Lin and Sontag 1991). Notably, as soon as simulations are introduced to check that theoretical conditions are true, the potential for vulnerabilities in the design (in the sense that the safety results may not hold) opens up. The more points that are checked within the stability region to ensure that the chosen α_3 satisfies **Eq. 2b** within that region, the greater the expectation one might have that it does everywhere (although an expectation is not a proof), but simulations are not as rigorous of a check as an analytic check. However, it may not always be possible to perform the checks analytically. Still, this is a part of the methodology that will need further improvements for designing safe systems under LEMPC and ultimately building to a cyberattack-resilient LEMPC design.

The first thing that we will check is that $\dot{V}$ is negative throughout the stability region that we plan to use so that we have reason to check if there is a negative definite upper-bounding function on $\dot{V}$ as required by **Eq. 2b**. Specifically, initially, a check was made that $\dot{V}$ was negative throughout Ω_ρ under the proposed $h(x)$ (saturated at the input bounds) for $\rho = 1{,}800$, by discretizing the state-space in the increments of 0.01 kmol/m^3 in C_A from 0 to 4 kmol/m^3, and in the increments of 1 K in T from 340 to 560 K. Since $\dot{V}$ was negative at the points tested, we suggest the function $\alpha_3(|x|) = a_3|x|^2$, with a_3 originally set to 100, and then, throughout the stability region, check whether $\dot{V}$ is less than the negative of this function. If it is not (implying that a_3 is too large), a_3 is changed to be equal to $-\dot{V}/|x|^2$ at the point where $\dot{V}$ was not less than or equal to $-\alpha_3(|x|)$. This results in $a_3 = 0.008\,22$; setting $a_3 = 0.008$ ensures that the inequality in **Eq. 2b** is satisfied at the points tested for this choice of $\alpha_3(|x|)$. Notably, a_3 is rounded down to obtain a suitable parameter, whereas the other parameters discussed below will be rounded up from the values returned by MATLAB because a_3 appears in a term that reflects a worst case when it is smaller, whereas the others appear in the terms that reflect the worst cases when they are larger.

The next function to be obtained is $\alpha_4(|x|)$. We again here guess a form for $\alpha_4(|x|)$ and then check whether **Eq. 2c** is satisfied at the points in the discretized stability region. Specifically, assuming that $\alpha_4(|x|) = a_4|x|^2$, we set a_4 initially to −100 and then update it to be $|\frac{\partial V}{\partial x}|/|x|^2$ whenever $|\frac{\partial V}{\partial x}| > \alpha_4(|x|)$. This gives that $a_4 = 8{,}156.72$ would work throughout the stability region with $\rho = 1{,}800$. We will choose $a_4 = 8{,}160$.

Next, the value of M_f is determined to satisfy **Eq. 5**. In this case, it is necessary to discretize not only the state-space within the stability region but also the input space and disturbance space. Furthermore, the upper bound on the magnitude of the disturbances will play a role in determining not only M_f but also whether the conditions of Proposition 2 and Proposition 6 and Theorem 6 are satisfied for the controller parameters. Again, the larger the value of ρ, the larger the value of M_f. To obtain M_f in this simulation, the state-space was discretized in the manner described above, and, in addition, the range of C_{A0} was discretized in the units of 0.5 kmol/m^3, while the range of Q was discretized in the units of 10^5 kJ/h. Furthermore, the disturbances used for this process had disturbance bounds of 2 kmol/m^3 h and 5 K/h for the disturbances added to the right-hand sides of **Eqs 40**, **41**, respectively. The disturbance space was therefore considered to go from −2 to 2 kmol/m^3 h in the units of 0.1 kmol/m^3 h for the disturbances added to the right-hand side of **Eq. 40** in deviation form and from −5 to 5 K/h in the increments of 0.5 K/h for the disturbances added to the right-hand side of **Eq. 41** in deviation form. M_f was originally set to 0, but then, it was changed to $|f(x,u,w)|$ at any of the discretized points where $|f(x, u, w)|$ was greater than the stored value of M_f. This results in a value of M_f within the stability region $\rho = 1{,}800$ of 4,465.75. The selected value for this simulation is 4,466.

L_x and L_w are the Lipschitz constants for f, as shown in **Eq. 4a**. To obtain these, first, L_x and L_w are obtained on their own by discretizing the state, input, and disturbance spaces, and finding the values that work when only the state is changed (for L_x) or when only the disturbances are changed (for L_w). Subsequently, it is checked that the resulting L_x and L_w satisfy **Eq. 4a** for the points in the discretized state-space. However, using the brute force method in this paper of checking many points (an aspect of this strategy that would scale poorly and therefore pose limitations for larger processes), the computation time can become many hours if the same discretization is used as was used above. Therefore, to obtain values for L_x and L_w more quickly, the discretization was made coarser; however, it should be understood that this also means that these parameter values (like the others above with other discretizations) are not necessarily the values that would be obtained with a finer discretization and therefore still leave the potential for safety vulnerabilities if the controller parameters are designed with these imperfect values of L_x and L_w. This also provides an insight into the challenges of using strategies like this for the safety-critical design of controllers, such as for the cyberattack-resilience extension.

Using a discretization of the input range of 1 kmol/m^3 in C_{A0}, of 10^5 kJ/h in Q, of 0.1 kmol/m^3 in C_A, of 1 K in T, and of 1 kmol/m^3 h for the disturbance added to the right-hand side of **Eq. 40** in a deviation form and of 1 K/h for the disturbance added to the right-hand side of **Eq. 41** in a deviation form, and then only looking at points in the stability region, the value of L_x was initialized at −1 and then reset to $|f(x, u, w) - f(x', u, w)|/|x - x'|$ whenever $|f(x, u, w) - f(x', u, w)| > L_x|x - x'|$ among the points checked. This resulted in a value of $L_x = 3{,}008.66$ being selected. A similar procedure for L_w gave $L_w = 1.00$. Then, a code that checks that **Eq. 4a** is satisfied at the points in the discretization with $L_x = 3{,}009$ and $L_w = 1.1$ was utilized, and the points in the discretization satisfied it.

Subsequently, it is necessary to calculate L_x' and L_w'. Using a similar strategy to that used in computing L_x and L_w, with the same discretization of the state, input, and disturbance spaces and only looking at points within the stability region, and setting the initial value of L_x' to −1 but updating it to $|\frac{\partial V(x)}{\partial x} f(x, u, w) - \frac{\partial V(x')}{\partial x} f(x', u, w)|/|x - x'|$ whenever $|\frac{\partial V(x)}{\partial x} f(x, u, w) - \frac{\partial V(x')}{\partial x} f(x', u, w)| > L_x'|x - x'|$ among the points checked, the value $L_x' = 439\,218.83$ results. Following a similar procedure for L_w', the value $L_w' = 3747.27$ results. Subsequently, it is checked that **Eq. 4b** is satisfied at the points checked with $L_x' = 439\,220$ and $L_w' = 3750$.

The final parameter to obtain is M_v in **Eq. 16**. This is obtained in a similar spirit to the methods above. Specifically, the range of C_A is discretized in the units of 0.01 kmol/m^3, while the range of T is discretized in the units of 1 K. M_v was originally set to 0. The points in this discretization in the stability region are examined. Subsequently, M_v is set to $(V(x) - V(\hat{x}) - \frac{a_4\rho}{\lambda_{\min}}|x - \hat{x}|)/(|x - \hat{x}|^2)$ if $(V(x) - V(\hat{x})) > \frac{a_4\rho}{\lambda_{\min}}|x - \hat{x}| + M_v(|x - \hat{x}|^2)$. The value of M_v after this algorithm was run was still 0. Therefore, M_v was set to 10^{-5}.

TABLE 3 | First set of parameters for CSTR model.

Parameter	Value
ρ	1,800
a_1	2.9
a_2	2,001
a_3	0.008
a_4	8,160
M_f	4,466
L_x	3,009
L_w	1.1
L_x'	439,200
L_w'	3,750
M_v	10^{-5}
θ_w	$\sqrt{29}$

The set of parameters obtained *via* these methods that is used in the first simulation is shown in **Table 3**. We note that many of these parameters were obtained within a given Ω_ρ, where if that region shrinks, it is possible that some values may change. To select the values of ρ_e', Δ, ρ_s, ϵ_w, and $\rho_{\min}$ that satisfy the conditions of Proposition 2 and Proposition 6 and Theorem 6, we consider formulating the following optimization problem:

$$\max_{\rho_e', \Delta, \rho_s, \bar{\epsilon}_w, \rho_{\min}} \rho_e' \tag{86a}$$

$$\text{s.t.} \quad \rho_e' - \rho + f_V\left(\bar{f}_W(\Delta)\right) \le 0 \tag{86b}$$

$$-\alpha_3\left(\alpha_2^{-1}(\rho_s)\right) + L_x' M_f \Delta + L_w' \theta_w + \bar{\epsilon}_w \le 0 \tag{86c}$$

$$\rho_s + L_x' M_f \Delta^2 + L_w' \theta_w \Delta - \rho_{\min} \le 0 \tag{86d}$$

$$\rho_{\min} - \rho_e' + 0.000\,01 \le 0 \tag{86e}$$

$$0 \le \rho_e' \le \rho \tag{86f}$$

$$0 \le \Delta \le 5 \tag{86g}$$

$$0 \le \rho_s \le \rho \tag{86h}$$

$$10^{-5} \le \bar{\epsilon}_w \le 10^{17} \tag{86i}$$

$$0 \le \rho_{\min} \le \rho \tag{86j}$$

In **Eq. 86**, $\bar{\epsilon}_w$ represents ϵ_w/Δ, so that the value of ϵ_w can be obtained from $\bar{\epsilon}_w\Delta$ after **Eq. 86** is solved. The objective function of **Eq. 86** was selected as ρ_e' to attempt to maximize the size of the region in which process economics is optimized under the constraint of **Eq. 9a**. **Equation 86b** was implemented as $\rho_e' - \rho + \frac{a_4\rho}{a_1}[\frac{L_w\theta_w}{L_x}(e^{L_x\Delta} - 1)] + M_v[\frac{L_w\theta_w}{L_x}(e^{L_x\Delta} - 1)]^2 \le 0$, and **Eq. 86c** was implemented as $-a_3\frac{\rho_s}{a_2} + L_x' M_f \Delta + L_w' \theta_w + \bar{\epsilon}_w \le 0$, in accordance with **Eqs 16**, **81b**, **82**, **83**. **Eq. 86d** was developed due to the fact that the closed-loop state may enter Ω_{ρ_s} under the operation of the LEMPC of **Eqs 8**, **9** with the constraint of **Eq. 9b** activated, where then:

$$\dot{V}(x(t)) \le -\alpha_3(|x(t_k)|) + \frac{\partial V(x(\tau))}{\partial x} f(x(\tau), u(t_k), w(\tau)) - \frac{\partial V(x(t_k))}{\partial x} f(x(t_k), u(t_k), 0) \tag{87}$$

for $t \in [t_k, t_{k+1})$, according to Eq. 18 in Heidarinejad et al. (2012a). In a worst case, $-\alpha_3(|x(t_k)|)$ is close to zero near the origin, so that it can be neglected. From the requirement of **Eq. 85**, $V(x(t_k)) + \dot{V}\Delta \le \rho_{\min}$ when $x(t_k) \in \Omega_{\rho_s}$. Substituting ρ_s and $\dot{V}$ from **Eq. 87** gives **Eq. 86d**. **Equation 86e** comes from the requirement that $\Omega_{\rho_{\min}} \subset \Omega_{\rho_e'}$. The bounds on the decision variables were set based on expectations of the values of the parameters and theoretical requirements. For example, because $\Omega_{\rho_{\min}} \subset \Omega_{\rho_e'} \subset \Omega_\rho$ and $\rho_{\min} > 0$, $\rho_e' > 0$, and $\rho > 0$ **Eqs 86f**, **86h**, **86j** were set (if the parameters ρ_e', ρ_s, or $\rho_{\min}$ were to equal zero in the result of **Eq. 86**, then our conclusion would be that the algorithm did not work properly). Δ should be positive (leading to the lower bound of 0 in **Eqs 86g**, where again if $\Delta = 0$, it would be considered that the result is problematic), and we expected it to be relatively small given the conditions of Proposition 2 and Proposition 6 and Theorem 6, so that an upper bound on Δ of 5 was selected in **Eq. 86g**, but this could be adjusted to be higher if desired.

TABLE 4 | Constraints of **Eq. 86** using the parameters of **Table 3**.

Equation number	Equation form
Equation 86b	$\rho_e' - 11\,770.90 + 9970.90e^{3009\Delta} + 3.88 \times 10^{-11}e^{6018\Delta} \leq 0$
Equation 86c	$-4.00 \times 10^{-6}\rho_s + 1961\,556\,520\Delta + 20\,194.37 + \bar{\epsilon}_w \leq 0$
Equation 86d	$\rho_s + 1961\ 556\ 520\Delta^2 + 20{,}194.37\Delta - \rho_{\min} \leq 0$
Equation 86e	$\rho_{\min} - \rho_e' + 0.000\,01 \leq 0$

Finally, due to a lack of knowledge of what value $\bar{\epsilon}_w$ should take besides that it should be positive, a large upper bound was provided to this parameter in **Eq. 86i**, with a lower bound enforcing that the parameter be positive. The lower bound of 10^{-5} was selected to prevent the parameter from decreasing all the way to zero, as it should be positive, but this lower bound could be adjusted. This optimization problem was solved in MATLAB using fmincon. From the initial guess $\rho_e' = 1$, $\Delta = 10^{-12}$, $\rho_s = 1$, $\bar{\epsilon}_w = 1$, and $\rho_{\min} = 1$, fmincon returned that the solution had converged to an infeasible point. To better understand the reason for the infeasibility and how to overcome it, the constraints can be analyzed one by one with the parameters shown in **Table 3**. Several of the constraints are shown in **Table 4**.

Our first task is to analyze what values of the decision variables might satisfy these constraints, particularly those of interest in applying LEMPC (e.g., larger sampling periods and values of ρ_e'). Considering first **Eq. 86b** in **Table 4**, we note that the value of Δ would need to be small due to the exponential terms in which Δ appears (for example, Δ of 10^{-5} h would enable **Eq. 86b** to be satisfied with ρ_e' at an example value in its allowable range (from **Eq. 86f**) of 1,000). However, moving to **Eq. 86c** in **Table 4**, we see that problems arise. First, we note that even if $\bar{\epsilon}_w$ takes its smallest value according to **Eq. 86i**, if $\Delta = 10^{-5}$ h, then ρ_s would need to be at least 9, 957, 459, 550, which is not less than ρ and is therefore not allowable. However, even if Δ was 0 (which is asymptotically the smallest value it could reach) and $\bar{\epsilon}_w$ was 10^{-5}, the term containing the noise bound θ_w would still cause the requirement on ρ_s to be that it be at least 5, 051, 116, 305, which again is much larger than ρ and therefore not allowable. This provides an indication that for the parameters of the LEMPC to provide guarantees for the selected values of ρ, V, h, $\alpha_1(\cdot)$, $\alpha_2(\cdot)$, $\alpha_3(\cdot)$, and $\alpha_4(\cdot)$, the value of θ_w needs to be small. In the following discussion, we will consider that it is 0 (no disturbances/plant-model mismatch).

If $\theta_w = 0$, then values of $\rho_e' = 1799$, $\rho_{\min} = 11$, $\rho_s = 10$, $\Delta = 10^{-15}$ h, and $\bar{\epsilon}_w = 10^{-5}$ satisfy the requirements of **Eqs 86b–86j**. However, this small sampling period would likely pose significant implementation challenges, particularly due to the need to execute an optimization problem every 10^{-15} h, and then it could also be challenging to simulate with these parameters (e.g., it could take a long time to simulate any substantial time length if 10^{-15} h was explicitly used as the time period). The problem with the sampling period size in this case is not only due to ρ_s being small; even if ρ_s was set to its maximum possible value of $\rho = 1{,}800$ from **Eq. 86h** in this case, $\bar{\epsilon}_w$ was set to its minimum value, and θ_w was set to 0, then **Eq. 86c** still indicates that Δ would need to be no more than 3.66×10^{-12} h. This motivates the question of what might happen to Δ if ρ was made smaller to affect some of the parameters in **Table 3**.

To investigate this, we can redo the procedure above for a different value of ρ that is smaller, to analyze the effects on the parameters of **Table 3** and also on the feasible space of **Eq. 86**. Selecting $\rho = 200$ (i.e., ρ is about an order of magnitude smaller than above) and neglecting disturbances, we note that a_1 and a_2 are fixed by P for the selected form of V, $\alpha_1(\cdot)$, and $\alpha_2(\cdot)$, so that if these are still "large" in the resulting problem, V, $\alpha_1(\cdot)$, or $\alpha_2(\cdot)$ would need to change to make an impact on these. Though ρ is smaller here, we do not update the discretization of the stability region used, as the values that are obtained from the above procedure provide a best case (i.e., additional points in the stability region can only make a_3 smaller, making it harder to find a larger Δ meeting **Eq. 86c**, and cannot make a_4, M_f, L_x, L_w, L_x', L_w', and M_v smaller, which can also make it harder to find a larger Δ meeting **Eq. 86c**). Therefore, we attempt to obtain a sense of whether changing the size of ρ allows Δ to be significantly larger than in the case with $\rho = 1{,}800$ with the coarser discretization.

The new parameters from the above procedure with $\rho = 200$ are shown in **Table 5**. With these updated parameters, **Eq. 86** gives a solution this time, specifically the solution in **Table 6**. The value of ρ_e' is maximized by driving it to its upper bound (since ρ_e' should be less than ρ, a constraint could be added in the future versions of this problem with a form similar to that in **Eq. 86e** but replacing $\rho_{\min}$ with ρ_e' and ρ_e with ρ, to enforce that $\Omega_{\rho_e'}$ is a strict subset of Ω_ρ). The value of Δ in **Table 6** is still incredibly small for process simulation. To check whether this is a fundamental limit of the parameters in **Table 5** or a function of the maximization of ρ_e' in **Eq. 86**, we can perform an analysis of the maximum possible value of Δ in **Eq. 86c**. For the parameters in **Table 5**, if ρ_s was its maximum possible value of $\rho = 200$ in **Eq. 86c** and $\bar{\epsilon}_w$ was its minimum possible value of 10^{-5}, then Δ in this equation would still need to be no larger than 2.245×10^{-10} h (again, a very small number).

The maximum possible value of Δ from this procedure from the case with $\rho = 200$ is about 2 orders of magnitude smaller than the maximum possible value of Δ for the case with $\rho = 1{,}800$; this begs the question of whether a further reduction of ρ may improve the situation (we also note that the discretization could play a role in this, which was not further explored in the preliminary analyses of this study). We could consider $\rho = 20$ for which the parameters obtained *via* the method above and **Eq. 86** are provided in **Tables 7**, **8**. In this case, although a_3 is increased compared to **Table 5** (at least among the points in the discretization used), ρ is smaller so that the maximum possible value of ρ_s is smaller and therefore the negative term in **Eq. 86c** does not become as large as would be desired to raise Δ. In this case, the maximum possible value of Δ from the procedure described is 3.82×10^{-10} h, which again is very small.

We see then that for the discretizations checked, decreasing the size of the stability region did not put the magnitude of Δ

TABLE 5 | Second set of parameters for CSTR model.

Parameter	Value
ρ	200
a_1	2.9
a_2	2,001
a_3	1.14
a_4	8,160
M_f	2,660
L_x	1,554
L_w	0
L_x'	190,800
L_w'	0
M_v	10^{-5}
θ_w	0

TABLE 6 | **Equation 86** parameters from the second set of parameters for CSTR model in **Table 5**.

Parameter	Value
ρ_e'	200.00
Δ	1.40×10^{-11}
ρ_s	37.49
$\bar{\epsilon}_w$	0.0071
$\rho_{\min}$	45.092

TABLE 7 | Third set of parameters for CSTR model.

Parameter	Value
ρ	20
a_1	2.9
a_2	2,001
a_3	11.15
a_4	5,601
M_f	2,294
L_x	1,221
L_w	0
L_x'	126,910
L_w'	0
M_v	10^{-5}
θ_w	0

TABLE 8 | **Equation 86** parameters from the third set of parameters for CSTR model in **Table 7**.

Parameter	Value
ρ_e'	20.00
Δ	7.09×10^{-11}
ρ_s	11.11
$\bar{\epsilon}_w$	0.021
$\rho_{\min}$	13.93

in a reasonable range for the selected for h, V, and α_i, $i = 1, 2, 3, 4$. This gives a greater insight into Remark 12, which indicated that it is necessary to select h and V such that reasonable parameters can be obtained. Future work could explore other functions h, V, and α_i, $i = 1, 2, 3, 4$ for this process to see whether there exists any that could result in more reasonable values of Δ or not. The results of this section also shed light on what changes could aid in making Δ larger (for example, it is seen above that a major reason why Δ is so small in each simulation is because a_3 is small compared to a_2 in each case and ρ_s is limited in magnitude by ρ, causing the only negative term in **Eq. 86c** to be small, and then since the terms that multiply Δ are large for the given shape of Ω_ρ, Δ must be small in each case to prevent the positive term containing Δ from overwhelming the negative term containing ρ_s and preventing $\bar{\epsilon}_w$ from being positive as required by **Eq. 86i**). Although these results have not focused directly on the cybersecurity of control systems, they give some indication of the challenges that would be faced in working toward developing the control parameters of a cyberattack-resilient LEMPC meeting the theory in this work. They indicate that meeting the theory requires better strategies than that used in this section for preventing vulnerabilities.

10 CONCLUSION

This work extended the control/detection strategies developed in Oyama and Durand (2020) to handle actuator attacks and cases where actuator and sensor attacks can occur simultaneously. For the event where multiple attacks are considered, several integrated control/detection frameworks that pair the detection strategies designed for single attack-type events were investigated. It was demonstrated that certain combinations of the detection strategies can be ineffective to flag both types of cyberattacks evaluated in this work, while others create a cyberattack-resilient structure that enables the detection of individual or simultaneous sensor and actuator attack types while ensuring safe operation even if undetected attacks occur. In particular, the pairing of Detection Strategies 1-A and 3-S and the pairing of Detection Strategies 2-A and 3-S were shown to be resilient against both types of cyberattacks. The major benefits of these methods are that multiple attack scenarios can be discovered, which adds a layer of protection, and closed-loop stability is guaranteed if an attack policy is not flagged by these two-piece structures. Finally, to characterize the fundamental nature of sensor and actuator attacks, we mathematically defined the concept of cyberattack discoverability in the context of process control and stealthy attack policies, which may provide insights for future detection strategy development. The potential practical challenges with designing LEMPCs meeting theoretical conditions, a precursor study for getting the parameters of cyberattack-resilient LEMPCs, elucidated some of the potential challenges with obtaining the parameters meeting the theory that could be addressed in future work.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusion of this article will be made available by the authors, without undue reservation. The codes generated for this study will be uploaded to: https://durand.eng.wayne.edu/PublicationResources/index.html.

AUTHOR CONTRIBUTIONS

HO worked on writing the manuscript and developing concepts and simulations for the manuscript. DM aided in developing simulations used for obtaining the results in the work and in writing portions of the work. KR aided in development of the actuator cyberattack-handling procedure. HD supervised the work and aided in writing and editing the manuscript and simulations and conceptualizing the methodology.

FUNDING

Financial support from the Air Force Office of Scientific Research (award number FA9550-19-1-0059), National Science Foundation CNS-1932026 and CBET-1839675, and Wayne State University is gratefully acknowledged.

ACKNOWLEDGMENTS

We would like to thank the reviewers, who provided tremendous help and insights in their comments that aided us greatly in developing the final version of this manuscript.

REFERENCES

Ahrens, J. H., and Khalil, H. K. (2009). High-gain Observers in the Presence of Measurement Noise: A Switched-Gain Approach. *Automatica* 45, 936–943. doi:10.1016/j.automatica.2008.11.012

Ani, U. P. D., He, H., and Tiwari, A. (2017). Review of Cybersecurity Issues in Industrial Critical Infrastructure: Manufacturing in Perspective. *J. Cyber Security Technol.* 1, 32–74. doi:10.1080/23742917.2016.1252211

Bemporad, A., Morari, M., Dua, V., and Pistikopoulos, E. N. (2002). The Explicit Linear Quadratic Regulator for Constrained Systems. *Automatica* 38, 3–20. doi:10.1016/s0005-1098(01)00174-1

Brunton, S. L., Proctor, J. L., and Kutz, J. N. (2016). Discovering Governing Equations from Data by Sparse Identification of Nonlinear Dynamical Systems. *Proc. Natl. Acad. Sci. USA* 113, 3932–3937. doi:10.1073/pnas.1517384113

Cárdenas, A. A., Amin, S., Lin, Z.-S., Huang, Y.-L., Huang, C.-Y., and Sastry, S. (2011). "Attacks against Process Control Systems: Risk Assessment, Detection, and Response," in Proceedings of the ACM Asia Conference on Computer & Communications Security, Hong Kong, China.

Chen, S., Wu, Z., and Christofides, P. D. (2020). A Cyber-Secure Control-Detector Architecture for Nonlinear Processes. *AIChE J.* 66, e16907. doi:10.1002/aic.16907

Davis, J., Edgar, T., Graybill, R., Korambath, P., Schott, B., Swink, D., et al. (2015). Smart Manufacturing. *Annu. Rev. Chem. Biomol. Eng.* 6, 141–160. doi:10.1146/annurev-chembioeng-061114-123255

Ding, D., Han, Q.-L., Ge, X., and Wang, J. (2020). Secure State Estimation and Control of Cyber-Physical Systems: A Survey. *IEEE Trans. Syst. Man, Cybernetics: Syst.* 51, 176–190.

Durand, H. (2018). A Nonlinear Systems Framework for Cyberattack Prevention for Chemical Process Control Systems. *Mathematics* 6, 44. doi:10.3390/math6090169

Durand, H., and Messina, D. (2020). "Enhancing Practical Tractability of Lyapunov-Based Economic Model Predictive Control," in Proceedings of the American Control Conference, Denver, Colorado, 2018–2023. doi:10.23919/acc45564.2020.9147880

Durand, H., and Wegener, M. (2020). Mitigating Safety Concerns and Profit/production Losses for Chemical Process Control Systems under Cyberattacks via Design/control Methods. *Mathematics* 8, 499. doi:10.3390/math8040499

Ellis, M., Durand, H., and Christofides, P. D. (2014a). A Tutorial Review of Economic Model Predictive Control Methods. *J. Process Control.* 24, 1156–1178. doi:10.1016/j.jprocont.2014.03.010

Ellis, M., Zhang, J., Liu, J., and Christofides, P. D. (2014b). Robust Moving Horizon Estimation Based Output Feedback Economic Model Predictive Control. *Syst. Control. Lett.* 68, 101–109. doi:10.1016/j.sysconle.2014.03.003

Griffith, D. W. (2018). "Advances in Nonlinear Model Predictive Control for Large-Scale Chemical Process Systems,". Ph.D. thesis (Pittsburgh, Pennsylvania: Carnegie Mellon University).

Heidarinejad, M., Liu, J., and Christofides, P. D. (2012a). Economic Model Predictive Control of Nonlinear Process Systems Using Lyapunov Techniques. *AIChE J.* 58, 855–870. doi:10.1002/aic.12672

Heidarinejad, M., Liu, J., and Christofides, P. D. (2012b). State-estimation-based Economic Model Predictive Control of Nonlinear Systems. *Syst. Control. Lett.* 61, 926–935. doi:10.1016/j.sysconle.2012.06.007

Khalil, H. K. (2002). *Nonlinear Systems*. Third edn. Upper Saddle River, New Jersey: Prentice-Hall.

Lao, L., Ellis, M., Durand, H., and Christofides, P. D. (2015). Real-time Preventive Sensor Maintenance Using Robust Moving Horizon Estimation and Economic Model Predictive Control. *AIChE J.* 61, 3374–3389. doi:10.1002/aic.14960

Lin, Y., and Sontag, E. D. (1991). A Universal Formula for Stabilization with Bounded Controls. *Syst. Control. Lett.* 16, 393–397. doi:10.1016/0167-6911(91)90111-q

Mhaskar, P., Liu, J., and Christofides, P. D. (2012). *Fault-tolerant Process Control: Methods and Applications*. Berlin, Germany: Springer Science & Business Media.

Muñoz de la Peña, D., and Christofides, P. D. (2008). Lyapunov-based Model Predictive Control of Nonlinear Systems Subject to Data Losses. *IEEE Trans. Automat. Contr.* 53, 2076–2089. doi:10.1109/tac.2008.929401

Narasimhan, S., El-Farra, N. H., and Ellis, M. J. (2021). Detectability-based Controller Design Screening for Processes under Multiplicative Cyberattacks. *AIChE J.* 68, e17430. doi:10.1002/aic.17430

Oyama, H., and Durand, H. (2020). Integrated Cyberattack Detection and Resilient Control Strategies Using Lyapunov-Based Economic Model Predictive Control. *AIChE J.* 66, e17084. doi:10.1002/aic.17084

Oyama, H., Rangan, K. K., and Durand, H. (2021). Handling of Stealthy Sensor and Actuator Cyberattacks on Evolving Nonlinear Process Systems. *J. Adv. Manufacturing Process.* 3, e10099. doi:10.1002/amp2.10099

Papachristodoulou, A., and Prajna, S. (2002). "On the Construction of Lyapunov Functions Using the Sum of Squares Decomposition," in *Proceedings of the IEEE Conference on Decision and Control* (Las Vegas, Nevada: IEEE), Vol. 3, 3482–3487.

Pasqualetti, F., Dörfler, F., and Bullo, F. (2013). Attack Detection and Identification in Cyber-Physical Systems. *IEEE Trans. Automat. Contr.* 58, 2715–2729. doi:10.1109/tac.2013.2266831

Qin, S. J., and Badgwell, T. A. (2003). A Survey of Industrial Model Predictive Control Technology. *Control. Eng. Pract.* 11, 733–764. doi:10.1016/s0967-0661(02)00186-7

Rangan, K. K., Oyama, H., and Durand, H. (2021). Integrated Cyberattack Detection and Handling for Nonlinear Systems with Evolving Process Dynamics under Lyapunov-Based Economic Model Predictive Control. *Chem. Eng. Res. Des.* 170, 147–179. doi:10.1016/j.cherd.2021.03.024

Rawlings, J. B., Angeli, D., and Bates, C. N. (2012). "Fundamentals of Economic Model Predictive Control," in Proceedings of the IEEE Conference on Decision and Control, Maui, Hawaii, 3851–3861. doi:10.1109/cdc.2012.6425822

Setola, R., Faramondi, L., Salzano, E., and Cozzani, V. (2019). An Overview of Cyber Attack to Industrial Control System. *Chem. Eng. Trans.* 77, 907–912.

Tuptuk, N., and Hailes, S. (2018). Security of Smart Manufacturing Systems. *J. Manufacturing Syst.* 47, 93–106. doi:10.1016/j.jmsy.2018.04.007

Wu, Z., Albalawi, F., Zhang, J., Zhang, Z., Durand, H., and Christofides, P. D. (2018). Detecting and Handling Cyber-Attacks in Model Predictive Control of Chemical Processes. *Mathematics* 6, 22. doi:10.3390/math6100173

Ye, D., and Luo, S. (2019). A Co-design Methodology for Cyber-Physical Systems under Actuator Fault and Cyber Attack. *J. Franklin Inst.* 356, 1856–1879. doi:10.1016/j.jfranklin.2019.01.009

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.