

6-2022

On-line Process Physics Tests via Lyapunov-based Economic Model Predictive Control and Simulation-Based Testing of Image-Based Process Control

Henrique Oyama

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

Akkarakaran Francis Leonard

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

Minhazur Rahman

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

Govanni Gjonaj

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

Michael Williamson

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

Follow this and additional works at: https://digitalcommons.wayne.edu/cems_eng_frp

See next page for additional authors



Part of the [Process Control and Systems Commons](#), and the [Systems Engineering Commons](#)

Recommended Citation

Oyama, Henrique; Leonard, Akkarakaran Francis; Rahman, Minhazur; Gjonaj, Govanni; Williamson, Michael; and Durand, Helen, "On-line Process Physics Tests via Lyapunov-based Economic Model Predictive Control and Simulation-Based Testing of Image-Based Process Control" (2022). *Chemical Engineering and Materials Science Faculty Research Publications*. 28.

https://digitalcommons.wayne.edu/cems_eng_frp/28

This Conference Proceeding is brought to you for free and open access by the Chemical Engineering and Materials Science at DigitalCommons@WayneState. It has been accepted for inclusion in Chemical Engineering and Materials Science Faculty Research Publications by an authorized administrator of DigitalCommons@WayneState.

Authors

Henrique Oyama, Akkarakaran Francis Leonard, Minhazur Rahman, Giovanni Gjonaj, Michael Williamson, and Helen Durand

On-line Process Physics Tests via Lyapunov-based Economic Model Predictive Control and Simulation-Based Testing of Image-Based Process Control

Henrique Oyama¹, Akkarakaran Francis Leonard¹, Minhazur Rahman¹, Giovanni Gjonaj¹, Michael Williamson¹, Helen Durand¹

Abstract—Next-generation manufacturing involves increasing use of automation and data to enhance process efficiency. An important question for the chemical process industries, as new process systems (e.g., intensified processes) and new data modalities (e.g., images) are integrated with traditional plant automation concepts, will be how to best evaluate alternative strategies for data-driven modeling and synthesizing process data. Two methods which could be used to aid in this are those which aid in testing data-based techniques on-line, and those which enable various data-based techniques to be assessed in simulation. In this work, we discuss two techniques in this domain which can be applied in the context of chemical process control, along with their benefits and limitations. The first is a method for testing data-driven modeling strategies on-line by postulating the experimental conditions which could reveal if a model is correct, and then attempting to collect data which could help to reveal this. The second strategy is a framework for testing image-based control algorithms via simulating both the generation of the images as well as the impacts of control on the resulting systems.

I. INTRODUCTION

Modern manufacturing relies on assured replicability, and selecting data-driven models appropriately, as well as testing control algorithms before deployment, is important for efficient process operation. To use model identification techniques, a suitable model structure that represents a dynamic system must be selected, which may include physics-based and/or empirical components [1], [2]. Experiment design [3] has been investigated to either carry out parameter estimation [4] or select model candidates [5]. The present paper develops a control-assisted online technique for model discrimination based on our previous work [6], which utilizes an advanced control law known as Lyapunov-based economic model predictive control (LEMPC) [7] to aid in evaluating whether proposed models may describe process physics. Additionally, as industrial control adapts to the needs of modern manufacturing, the ability to move systems traditionally tested online into virtual testing simulations becomes increasingly important. This paper focuses on one of those systems - image-based control (IBC). Although image-based closed-loop systems have been used for real

systems involving camera sensors [8], [9], simulation-based methods for evaluating image-based control designs require attention for chemical processes. Moving towards simulating IBC systems, 3D creation software such as Blender, which has an embedded Python interpreter, may help with creating animated simulations that allow evaluation of image-based control integrated with processes for industrial applications. In this work, we investigate a closed-loop simulation of an IBC system developed in Blender to demonstrate benefits and limitations of 3D creation platforms for studying image-based control designs and their implications for a process.

II. PRELIMINARIES

A. Notation

R corresponds to the set of real numbers. 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 $\alpha : [0, a) \rightarrow [0, \infty)$ 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 $\Omega_\rho := \{x \in R^n : V(x) \leq \rho\}$.

B. Class of Systems

The class of nonlinear systems considered is the following:

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

where $x \in X \subset R^n$ and $u \in U \subset R^m$ are the state and input vectors, respectively, in deviation variable form from the steady-state (x_s) and steady-state input of the system (u_s); $w \in W \subset R^z$ ($W := \{w \in R^z \mid |w| \leq \theta, \theta > 0\}$) is the disturbance vector and f is locally Lipschitz on $X \times U \times W$. We consider that the “nominal” system of Eq. 1 ($w \equiv 0$) satisfies $f(0, 0, 0) = 0$ and is stabilizable such that there exists an asymptotically stabilizing feedback control law $h(x)$, a sufficiently smooth Lyapunov function $V(x)$, and class \mathcal{K} functions $\alpha_i(\cdot)$, $i = 1, 2, 3, 4$, where:

$$\alpha_1(|x|) \leq V(x) \leq \alpha_2(|x|) \quad (2a)$$

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

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

$$h(x) \in U \quad (2d)$$

¹Henrique Oyama, Akkarakaran Francis Leonard, Minhazur Rahman, Giovanni Gjonaj, Michael Williamson, and Helen Durand (helen.durand@wayne.edu) are with Department of Chemical Engineering and Materials Science, Wayne State University, 5050 Anthony Wayne Drive, Detroit, MI

$\forall x \in D \subset R^n$ (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 $x \in X$, $\forall x \in \Omega_\rho$. We consider that state measurements are available continuously, but are only used by a controller at discrete sampling times. Because f is a locally Lipschitz function of its arguments, we can write the following $\forall x_1, x_2 \in \Omega_\rho, u \in U$, $w \in W$, and L_x, L'_x, L_w, L'_w , and M_f as positive constants:

$$|f(x_1, u, w) - f(x_2, u, 0)| \leq L_x|x_1 - x_2| + L_w|w| \quad (3a)$$

$$\left| \frac{\partial V(x_1)}{\partial x} f(x_1, u, w) - \frac{\partial V(x_2)}{\partial x} f(x_2, u, 0) \right| \leq L'_x|x_1 - x_2| + L'_w|w| \quad (3b)$$

$$|f(x, u, w)| \leq M_f \quad (4)$$

We consider that an i -th empirical model is represented by $\hat{x}_i = f_{NL,i}(x_i(t), u_i(t))$ where Eqs. 2a-4 hold with respect to the empirical model (i.e., replace α_i , $i = 1, \dots, 4$ by $\hat{\alpha}_i$, replace M_f by $M_{f,i}$, replace V by \hat{V}_i , replace h by $h_{NL,i}$, replace L_x, L_w, L'_x and L'_w with $\hat{L}_{x,i}, \hat{L}_{w,i}, \hat{L}'_{x,i}$ and $\hat{L}'_{w,i}$, and replace ρ with ρ_i). u_i represents the input vector in deviation form from the steady-state input of the i -th model. We assume that $x = x_s$ is a steady-state for both f and $f_{NL,i}$, with different steady-state inputs u_{is} for the empirical models compared to u_s for the actual system.

C. Lyapunov-Based Economic Model Predictive Control

LEMPC [7] computes control actions via:

$$\min_{u(t) \in S(\Delta)} \int_{t_k}^{t_k+N} L_e(\tilde{x}(\tau), u(\tau)) d\tau \quad (5a)$$

$$\text{s.t. } \dot{\tilde{x}}(t) = f(\tilde{x}(t), u(t), 0) \quad (5b)$$

$$\tilde{x}(t_k) = x(t_k) \quad (5c)$$

$$\tilde{x}(t) \in X, \forall t \in [t_k, t_k+N) \quad (5d)$$

$$u(t) \in U, \forall t \in [t_k, t_k+N) \quad (5e)$$

$$V(\tilde{x}(t)) \leq \rho_e, \forall t \in [t_k, t_k+N), \quad (5f)$$

$$\begin{aligned} & \text{if } x(t_k) \in \Omega_{\rho_e} \\ & \frac{\partial V(\tilde{x}(t_k))}{\partial x} f(\tilde{x}(t_k), u(t_k), 0) \\ & \leq \frac{\partial V(\tilde{x}(t_k))}{\partial x} f(\tilde{x}(t_k), h(x(t_k)), 0), \\ & \text{if } x(t_k) \in \Omega_\rho / \Omega_{\rho_e} \end{aligned} \quad (5g)$$

where $u(t)$ is a piecewise-constant input trajectory with N pieces, where each piece is held constant for a sampling period with time length Δ , and N is the prediction horizon. The economics-based stage cost L_e of Eq. 5a is evaluated throughout the prediction horizon based on the future predictions of the process state \tilde{x} from the model of Eq. 5b started from the state measurement at t_k (Eq. 5c). The process constraints of Eqs. 5d-5e are state and input constraints, respectively. $\Omega_{\rho_e} \subset \Omega_\rho$ is a subset of the stability region that makes Ω_ρ forward invariant under the LEMPC of Eq. 5.

III. AUTOMATED ONLINE CONTROL-ASSISTED MODEL STRUCTURE DISCRIMINATION USING LEMPC

Prior work [10] in our group has proposed control-assisted designs using LEMPC to seek to collect desired data online for selecting model structures while ensuring closed-loop stability in a safe region of operation. However, [10] did not provide a way to automatically choose what desired data should be. Subsequently, in [6], we defined desired data as the information that could discriminate between rival models and sought to collect this data using the LEMPC objective function (with an economics-based stage cost and/or a stage cost rewarding differences between state predictions from different models under the control actions). This strategy ensures safety while gathering data for trying to discriminate between model candidates, but is less targeted in the data that it collects than might be desired from the perspective of performing experimentation online to determine an appropriate model for an unknown physical system. Scientific experiments are often performed to either verify or refute a hypothesis. This implies that another way for defining desired data for discriminating between model candidates could be data that is able to refute or satisfy a hypothesis. The flexibility of LEMPC allows it to be designed to drive the process state to a neighborhood of an operating point when the plant/model mismatch is sufficiently small (i.e., $|f(x, u, w) - f_{NL,i}(x, u)| \leq M_{err,i}$, where $M_{err,i}$ indicates the plant/model mismatch for all $x \in \Omega_{\rho_{|M_c|,1}}$, $u \in U$, and $w \in W$ and would be bounded by a stability/feasibility analysis of LEMPC if guarantees were to be made such as the LEMPC driving the closed-loop state toward an operating point). Then, if an LEMPC is used to attempt to drive the closed-loop state toward a desired operating point but data obtained indicates that the closed-loop state was not driven near that point, this may indicate that the model used in computing the control actions was not accurate enough to probe state-space and could be discarded. This procedure then serves as online experimentation for discriminating between models and potentially better understanding the process physics. An LEMPC operated in this fashion can use a constraint of the form of Eq. 5g to attempt to force the process state toward desired data and “check” different model candidates. In the next section, we present this control-assisted scheme.

A. Control-Assisted Online Model Structure Discrimination using LEMPC: Formulation

To achieve the data-gathering goals described above and attempt to discriminate between models, the set of $|M_c|$ model candidates must first be developed, where $|M_c|$ represents the cardinality of a set of process models M_c . This set will be assumed to contain a model where $M_{err,i}$ is sufficiently small in the sense that the closed-loop state under the LEMPC to be developed could be driven toward a neighborhood of a desired operating point when the i -th model is used in a constraint with the form of Eq. 5g. We also assume that the models in the set are ordered such that their stability regions are nested. This is done to ensure that the closed-loop

state does not leave the stability region of the sufficiently accurate model when the constraints with the forms of Eqs. 5f-5g utilize a model that is not sufficiently accurate. We will denote the stability region of the sufficiently accurate model as $\Omega_{\rho_{a,1}}$, where $a \in \{1, \dots, |M_c|\}$, but this region corresponding to the a -th model is not known until the data-gathering process is concluded. Furthermore, the proposed LEMPC will discriminate between models by seeking to drive the closed-loop state toward different operating data points. We assume that in the time of operation, p such points are selected to be tracked, and we denote them by $x_{s,j}$, $j = 2, \dots, p$ ($x_{s,1}$ represents the operating steady-state when no non-routine operating data is being gathered).

The operating strategy then consists of progressively attempting to drive the closed-loop state toward each of the $x_{s,j}$, $j = 2, \dots, p$, by switching to an LEMPC with Eq. 5g based on the model in M_c with the smallest stability region (corresponding to $i = 1$) and with the models and all constraints rewritten to consider the equilibrium to be at a steady-state $x_{s,j}$. When the closed-loop state does not approach $x_{s,j}$ in the subsequent operating period, the model in M_c with the smallest stability region can be discarded from that set, so that now that which previously had the second largest stability region becomes that with the smallest stability region (a metric on the ‘‘approach’’ of the closed-loop state to $x_{s,j}$ and a threshold ϵ_D on this metric that leads to models being discarded can be a design decision). We can therefore consider that when the LEMPC operates in a data-gathering mode (i.e., seeking to track $x_{s,j}$), it is probing whether the model candidates in M_c are sufficiently accurate according to the selected approach metric and threshold. However, for the majority of the time of operation, the closed-loop system is operated under a ‘‘baseline’’ 1-LEMPC to only optimize economics around $x_{s,1}$. The 1-LEMPC formulation is as follows:

$$\begin{aligned} & \min_{u_{1,1}(\cdot) \in S(\Delta)} \int_{t_k}^{t_{k+N}} \left[\sum_{i=1}^{|M_c|} L_e(\tilde{x}_{i,1}(\tau), u_{1,1}(\tau)) \right] d\tau \\ \text{s.t. } & \dot{\tilde{x}}_{i,1}(t) = f_{NL,i,1}(\tilde{x}_{i,1}(t), u_{i,1}(t)), i = 1, \dots, |M_c| \quad (6a) \\ & \tilde{x}_{i,1}(t_k) = x(t_k), i = 1, \dots, |M_c| \quad (6b) \\ & \tilde{x}_{1,1}(t) \in X, \forall t \in [t_k, t_{k+N}] \quad (6c) \\ & u_{1,1}(t) \in U_{1,1}, \forall t \in [t_k, t_{k+N}] \quad (6d) \\ & \hat{V}_{1,1}(\tilde{x}_{1,1}(t)) \leq \rho_{e,1,1}, \forall t \in [t_k, t_{k+N}] \\ & \quad \text{if } \hat{V}_{1,1}(\tilde{x}_{1,1}(t_k)) \leq \rho_{e,1,1} \quad (6e) \\ & \frac{\partial \hat{V}_{1,1}(\tilde{x}_{1,1}(t_k))}{\partial \tilde{x}_{1,1}} f_{NL,1,1}(\tilde{x}_{1,1}(t_k), u_{1,1}(t_k)) \leq \\ & \frac{\partial \hat{V}_{1,1}(\tilde{x}_{1,1}(t_k))}{\partial \tilde{x}_{1,1}} f_{NL,1,1}(\tilde{x}_{1,1}(t_k), h_{NL,1,1}(\tilde{x}_{1,1}(t_k))) \\ & \quad \text{if } \hat{V}_{1,1}(\tilde{x}_{1,1}(t_k)) > \rho_{e,1,1} \quad (6f) \end{aligned}$$

where L_e is the EMPC objective function, $\tilde{x}_{i,1}$ is the state prediction in deviation variable form from $x_{s,1}$ based on the i -th model candidate, and $x(t_k)$ is the state measurement at

t_k (with slight abuse of notation, this is used in Eq. 6b to represent the deviation form from the steady-state for the i -th model). In Eq. 6, many terms have two subscripts; the first refers to the process model under consideration, and the second refers to the j -th steady-state, $j = 1, \dots, p$. The controller of Eq. 6 is used if $t_{e,j-1} \leq t < t_{s,j}$, $j = 2, \dots, p$, where $t_{e,1} = 0$; $t_{s,j}$ is defined as the switching time when the LEMPC changes to drive the closed-loop state to the j -th desired data point ($j > 1$), and $t_{e,j}$ is the time at which the control law switches back to the 1-LEMPC.

At $t_{s,j}$, the LEMPC formulation is switched from the 1-LEMPC to that associated with $x_{s,j}$ to drive the closed-loop state to a neighborhood of $x_{s,j}$. The j -th LEMPC, $j > 1$, which is used for $t \in [t_{s,j}, t_{e,j})$, is formulated as follows:

$$\begin{aligned} & \min_{u_{1,j}(\cdot) \in S(\Delta)} \int_{t_k}^{t_{k+N}} \left[\sum_{i=1}^{|M_c|} L_e(\tilde{x}_{i,j}(\tau), u_{1,j}(\tau)) \right] d\tau \\ \text{s.t. } & \dot{\tilde{x}}_{i,j}(t) = f_{NL,i,j}(\tilde{x}_{i,j}(t), u_{i,j}(t)), i = 1, \dots, |M_c| \quad (7a) \\ & \tilde{x}_{i,j}(t_k) = x(t_k), i = 1, \dots, |M_c| \quad (7b) \\ & \tilde{x}_{1,j}(t) \in X, \forall t \in [t_k, t_{k+N}] \quad (7c) \\ & u_{1,j}(t) \in U_{1,j}, \forall t \in [t_k, t_{k+N}] \quad (7d) \\ & \frac{\partial \hat{V}_{1,j}(\tilde{x}_{1,j}(t_k))}{\partial \tilde{x}_{1,j}} f_{NL,1,j}(\tilde{x}_{1,j}(t_k), u_{1,j}(t_k)) \leq \\ & \frac{\partial \hat{V}_{1,j}(\tilde{x}_{1,j}(t_k))}{\partial \tilde{x}_{1,j}} f_{NL,1,j}(\tilde{x}_{1,j}(t_k), h_{NL,1,j}(\tilde{x}_{1,j}(t_k))) \quad (7e) \end{aligned}$$

where $\tilde{x}_{i,j}(t_k)$ represents the state measurement for the i -th model in deviation variable form from the j -th steady-state.

One of the benefits of this strategy is that it provides a means for manufacturers to have a system attempt to discover its own physics and to provide data over time which can help to uncover aspects of the physics, as opposed to only routine operating data. As concepts in learning from data gain prominence, the potential for a process to gather non-routine data in a manner that is expected to be revealing has potential to be helpful in developing data-based models on-line. However, if it is done while the process is on-line, it is necessary to consider the impact of this on profits and to attempt to reduce that impact. For example, one could explore activating the j -th LEMPC when only a few sampling periods would be required to drive the state to a neighborhood of $x_{s,j}$ based on $x(t_k)$ (where the number of sampling periods allowable will depend on the timescale on which the process profit metric evolves). Because the decision to switch to a j -th LEMPC may depend on the location of the closed-loop state in state-space and how moving toward desired data could impact profits, $t_{s,j}$ and $t_{e,j}$ would be expected to be determined online (further detail on such an implementation strategy is presented in the next section, where the impacts of selecting a certain $x_{s,j}$ on the closed-loop stability/feasibility guarantees of the j -th LEMPC also are taken into account in selecting $t_{s,j}$ by choosing to switch to the j -th LEMPC only when the

closed-loop state is within a stability region containing the new steady-state). Limitations of this LEMPC-based method for gathering non-routine operating data for aiding in model discrimination include that the set of M_c models must be developed *a priori* and should include a sufficiently accurate model in the set, and that requiring the nesting of stability regions has a potential to be conservative.

B. Control-Assisted Online Model Structure Discrimination using LEMPC: Implementation Strategy

Assuming that a reasonably accurate model is used by the proposed LEMPC design, the implementation strategy below includes a region $\Omega_{\rho_{e,1,j}}$, selected such that if the actual state is in $\Omega_{\rho_{e,1,j}} \subset \Omega_{\rho_{a,1}}$, under sufficient conditions, then the closed-loop state is maintained in $\Omega_{\rho_{a,1}}$ for $t \geq 0$. Information may be gathered automatically as follows:

- 1) At t_0 , set an index $\zeta = 0$ and set $t_{e,j} = 0$. Go to Step 2.
- 2) At the sampling time t_k , if $\zeta = 1$ and $t_k < t_{e,j}$, go to Step 6. If $\zeta = 1$ and $t_k = t_{e,j}$, set $t_{e,j} = 0$ and $\zeta = 0$, and go to Step 3. Otherwise, evaluate if process data is desired to be collected. If so, set $\zeta = 1$ and go to Step 4. Otherwise, set $\zeta = 0$ and go to Step 3.
- 3) The 1-LEMPC of Eq. 6 is activated and receives the state measurement $x(t_k)$. Go to Step 5.
- 4) Evaluate if a j -th steady-state ($j > 1$) corresponding to the desired information can be generated such that: 1) $\Omega_{\rho_{1,j}} \subset \Omega_{\rho_{i,j}}$, $i = 2, \dots, |M_c|$, around this steady-state; 2) $x_{s,j}$ has a steady-state input within the input bounds for all $|M_c|$ models; 3) $\Omega_{\rho_{e,1,j}}$ contains the state measurement $x(t_k)$ and $x_{s,j}$; and 4) $\Omega_{\rho_{|M_c|,j}} \subset \Omega_{\rho_{e,1,1}}$. If this is possible, set $t_k = t_{s,j}$ and select $t_{e,j}$ to be sufficiently long to drive the closed-loop state to a neighborhood of the desired information, and go to Step 6. If these conditions cannot be satisfied, go to Step 3 and set $\zeta = 0$.
- 5) If $x(t_k) \in \Omega_{\rho_{e,1,1}}$, go to Step 5a. Else, go to Step 5b.
 - a) Compute a control action for the subsequent sampling period with Eq. 6e of the 1-LEMPC activated. Go to Step 7.
 - b) Compute a control action for the subsequent sampling period with Eq. 6f of the 1-LEMPC activated. Go to Step 7.
- 6) The j -LEMPC of Eq. 7 is activated and receives the state measurement $x(t_k)$. The controller computes control actions to drive the closed-loop state to the desired information $x_{s,j}$ until $t_{e,j}$. Go to Step 7.
- 7) If $\tilde{x}_{i,j}$ exits the stability region of the i -th model candidate at any $t \in [t_k, t_{k+1})$, or if $|\tilde{x}_{i,j}(t_{k+1}) - x(t_{k+1})| \geq \epsilon_D$, then the i -th model is discarded from the set M_c and the set M_c is updated to have the models renumbered such that $i \leftarrow i + 1$, $i = 1, 2, \dots, |M_c| - 1$. Go to Step 8.
- 8) Go to Step 2 ($k \leftarrow k + 1$).

1) *Control-Assisted Online Model Discrimination using LEMPC: Stability Analysis:* In this section, we demonstrate

that the implementation strategy of Section III-B maintains the closed-loop state within $\Omega_{\rho_{a,1}}$ at all times.

Theorem 1: Consider the closed-loop system of Eq. 1 under the implementation strategy of Section III-B, where $h_{NL,i,j}(\cdot)$ used in the LEMPC's of Eqs. 6-7 for any i -th model in the set M_c and $j = 1, \dots, p$ meets the inequalities in Eqs. 2a-2d with respect to the i -th empirical model candidate. Let $\epsilon_{w,i,j} > 0$, $\epsilon'_{w,a,1} > 0$, $\bar{\epsilon}'_{w,a,j} > 0$, $\bar{L}'_{x,j} > 0$, $\bar{L}'_{w,j} > 0$, $\Delta > 0$, and $N \geq 1$. At every sampling time, let $\Omega_{\rho_{1,j}} \subset \Omega_{\rho_{i,j}} \subset X$ and $\Omega_{\rho_{|M_c|,j}} \subset \Omega_{\rho_{e,1,1}}$, where $\rho_{1,j} = \min\{\rho_{i,j}\}$, $\rho_{i,j} > \rho_{e,i,j} > \rho_{\min,i,j} > \rho_{s,i,j}$, for $i = 1, 2, \dots, |M_c|$ and $j = 1, \dots, p$, and $\rho_{e,i,j} > \rho_{i-1,j}$ for $i = 2, \dots, |M_c|$, satisfy:

$$-\hat{\alpha}_{3,i,j}(\hat{\alpha}_{2,i,j}^{-1}(\rho_{s,i,j})) + \hat{L}'_{x,i,j} \hat{M}_{f,i} \Delta \leq -\epsilon_{w,i,j} / \Delta, \quad (8)$$

$$i = 1, \dots, |M_c|$$

$$\left| \frac{\partial \hat{V}_{a,j}(x_1)}{\partial x} f_j(x_1, u, w) - \frac{\partial \hat{V}_{a,j}(x_2)}{\partial x} f_j(x_2, u, 0) \right| \leq \bar{L}'_{x,j} |x_1 - x_2| + \bar{L}'_{w,j} |w|, \quad \forall x \in \Omega_{\rho_{|M_c|,1}} \quad (9)$$

$$\left| \frac{\partial \hat{V}_{a,1}(x(t))}{\partial x} - \frac{\partial V_1(x(t))}{\partial x} \right| \leq M_{g,a,1}, \quad (10)$$

$$M_{g,a,1} > 0, \quad \forall x \in \Omega_{\rho_{|M_c|,1}}$$

$$-\hat{\alpha}_{3,a,1}(\hat{\alpha}_{2,a,1}^{-1}(\rho_{e,a,1})) + \hat{\alpha}_{4,a,1}(\hat{\alpha}_{1,a,1}^{-1}(\rho_{a,1})) M_{err,a} + L'_x M_f \Delta + L'_w \theta + 2M_{g,a,1} M_f \leq -\epsilon'_{w,a,1} / \Delta \quad (11)$$

$$-\hat{\alpha}_{3,a,j}(\hat{\alpha}_{2,a,j}^{-1}(\rho_{s,a,j})) + \hat{\alpha}_{4,a,j}(\hat{\alpha}_{1,a,j}^{-1}(\rho_{a,j})) M_{err,a} + \bar{L}'_{x,j} M_f \Delta + \bar{L}'_{w,j} \theta \leq -\bar{\epsilon}'_{w,a,j} / \Delta \quad (12)$$

$$\rho_{e,a,1} + f_{V,a,1}(f_{W,a,1}(\Delta)) \leq \rho_{a,1} \quad (13)$$

$$\rho_{e,i,j} \geq \max\{\hat{V}_{i,j}(x(t)) : x(t_k) \in \Omega_{\rho_{i-1,j}}, w \in W, t \in [t_k, t_{k+1}), u_{1,j} \in U\}, \quad i = 2, \dots, |M_c| \quad (14)$$

$$\rho_{i,j} \geq \max\{\hat{V}_{i,j}(x(t)) : x(t_k) \in \Omega_{\rho_{i-1,j}}, w \in W, t \in [t_k, t_{k+1}), u_{1,j} \in U\}, \quad i = 2, \dots, |M_c| \quad (15)$$

$$\rho_{\min,i,j} = \max\{\hat{V}_{i,j}(x(t)) : x(t_k) \in \Omega_{\rho_{s,i,j}}, w \in W, t \in [t_k, t_{k+1}), u_{1,j} \in U\}, \quad i = 1, \dots, |M_c| \quad (16)$$

$$\rho_{\min,i,j} = \max\{\hat{V}_{i,j}(x_{i,j}(t)) : x_{i,j}(t_k) \in \Omega_{\rho_{s,i,j}}, w \in W, t \in [t_k, t_{k+1}), u_{1,j} \in U\}, \quad i = 1, \dots, |M_c| \quad (17)$$

where $f_{W,i}(\tau) := \left(\frac{L_w \theta + M_{err,i}}{L_x} \right) e^{(L_x \tau - 1)}$ and $f_{V,i,1}(s) := \hat{\alpha}_{4,i,1}(\hat{\alpha}_{1,i,1}^{-1}(\rho_{i,1}))s + M_{v,i,1}s^2$, $M_{v,i,1} > 0$. If $\tilde{x}(t_0) = x(t_0) \in \Omega_{\rho_{e,1,1}}$, then the closed-loop state is maintained in $\Omega_{\rho_{a,1}}$ for $t \geq 0$.

Proof: This proof consists of multiple parts. In the first, recursive feasibility of the 1-LEMPC and the j -LEMPC is demonstrated. In the second, third, and fourth, we build to demonstrate that the closed-loop state is maintained in $\Omega_{\rho_{a,1}}$.

Part 1: Recursive feasibility of the 1-LEMPC and j -LEMPC holds because $h_{NL,1,j}$ satisfies Eqs. 6e-6f and 7e and maintains the state prediction in $\Omega_{\rho_{1,j}}$ if Eqs. 8 and 17 are satisfied [10]. Also since $\Omega_{\rho_{i,j}} \subset X$, $i = 1, 2, \dots, |M_c|$,

$j = 1, \dots, p$, Eqs. 6c and 7c are met under $h_{NL,1,j}$. Finally, $h_{NL,1,j}$ satisfies Eqs. 6d and 7d by Eq. 2d applied for the empirical model.

Part 2: Under the conditions in Theorem 1, [6] demonstrates that when the 1-LEMPC is utilized and $x(t_k) \in \Omega_{\rho_{a,1}}$ (as enforced by Step 7 of the implementation strategy), then the closed-loop state remains in $\Omega_{\rho_{a,1}}$ until t_{k+1} .

Part 3: We now demonstrate that the j -LEMPC maintains the closed-loop state in $\Omega_{\rho_{a,1}}$. By Steps 4 and 7 of the implementation strategy, when $t_k \in [t_{s,j}, t_{e,j})$, $x(t_k) \in \Omega_{\rho_{1,j}}$. In this case, either $\Omega_{\rho_{a,j}} = \Omega_{\rho_{1,j}}$ or $\Omega_{\rho_{i,j}}$, $i > 1$, is $\Omega_{\rho_{a,j}}$. The implementation strategy develops a new steady-state $x_{s,j}$ such that the stability region $\Omega_{\rho_{1,j}}$ for the $i = 1$ empirical model around the new steady-state contains $x(t_k)$ and $x_{s,j}$. This stability region should also be developed such that it meets the assumptions in Eqs. 2a-2d for the $i = 1$ empirical model formulated with respect to $x_{s,j}$. The stability regions for the other models, $j > 1$ are required to contain $\Omega_{\rho_{1,j}}$ and be fully contained in $\Omega_{\rho_{e,1}}$. In this case, Eq. 7e is the constraint utilized. If $\rho_{a,j} = \rho_{1,j}$, then if $x(t_k) \in \Omega_{\rho_{a,j}}/\Omega_{\rho_{s,a,j}}$, Eqs. 7e and 2b give:

$$\begin{aligned} & \frac{\partial \hat{V}_{1,j}(x(t_k))}{\partial x} f_{NL,1,j}(x(t_k), u_{1,j}(t_k)) \\ & \leq \frac{\partial \hat{V}_{1,j}(x(t_k))}{\partial x} f_{NL,1,j}(x(t_k), h_{NL,1,j}(x(t_k))) \\ & \leq -\hat{\alpha}_{3,a,j}(|x(t_k)|) \end{aligned} \quad (18)$$

Defining $\hat{V}_{1,j} = \frac{\partial \hat{V}_{1,j}(x_j(t))}{\partial x} f_j(x_j(t), u_{1,j}(t_k), w(t))$, adding and subtracting both $\frac{\partial \hat{V}_{1,j}(x(t_k))}{\partial x} f_{NL,1,j}(x(t_k), u_{1,j}(t_k))$ and $\frac{\partial \hat{V}_{1,j}(x(t_k))}{\partial x} f_j(x(t_k), u_{1,j}(t_k), 0)$ to $\hat{V}_{1,j}$ (with slight abuse of the notation $u_{1,j}(t_k)$ to stand for the same input in appropriate deviation form for the model it is in), and using Eq. 18, Eq. 3b applied to the model of Eq. 1 in deviation form from $x_{s,j}$, the definition of $M_{err,i}$, the bound on w , Eq. 4, Eq. 9, and Eq. 2a and 2c, we obtain that $\hat{V}_{1,j}$ is bounded by the left-hand side of Eq. 12. Therefore, if Eq. 12 holds, $\hat{V}_{1,j}$ decreases over a sampling period when $x(t_k) \in \Omega_{\rho_{a,j}}/\Omega_{\rho_{s,a,j}}$ if $\rho_{a,j} = \rho_{1,j}$. This ensures that $x(t_{k+1}) \in \Omega_{\rho_{1,j}} \subset \Omega_{\rho_{a,1}}$. If instead $x(t_k) \in \Omega_{\rho_{s,a,j}}$ and $\rho_{a,j} = \rho_{1,j}$, then Eq. 16 ensures that $x(t_{k+1}) \in \Omega_{\rho_{min,a,j}} \subset \Omega_{\rho_{a,j}} \subset \Omega_{\rho_{a,1}}$. If $\rho_{a,j} \neq \rho_{1,j}$, Eq. 15 ensures that $x(t_{k+1}) \in \Omega_{\rho_{a,j}} \subset \Omega_{\rho_{a,1}}$. Thus, with this implementation strategy, the closed-loop state is maintained in $\Omega_{\rho_{a,1}}$ from t_k to t_{k+1} .

Part 4. $x(t_0) \in \Omega_{\rho_{e,1,1}}$ by the statement of the theorem. At t_0 , either the 1-LEMPC or the j -LEMPC is utilized. If the 1-LEMPC is used, then from Part 2, $x(t) \in \Omega_{\rho_{a,1}}$, for $t \in [t_k, t_{k+1})$. If the j -LEMPC is used, then from Part 3, $x(t) \in \Omega_{\rho_{a,1}}$, for $t \in [t_k, t_{k+1})$. Applying Parts 2 and 3 recursively shows that the closed-loop state is maintained in $\Omega_{\rho_{a,1}}$ at all times. ■

IV. TEST FRAMEWORK FOR IMAGE-BASED CONTROL USING BLENDER

The prior section discussed a method for on-line testing of data-driven models. In this section, we develop a framework

for *a priori* testing of process control systems based on images. As image data becomes more important in the context of Industry 4.0, it is desirable to have a means for testing how an automated system might interact with image-based control. Rather than having an experimental system to test such studies for chemical processes, we explore the use of the 3D creation suite Blender [11] to better understand what the potential benefits and limitations of the use of this type of software might be for such a purpose.

A. Level Control Example: Image-based Level Control using Classical Controller with a Fixed Camera Sensor

In this section, a level control example is used to explore how the image-based control simulation described above might work. The process is represented by Eq. 19:

$$\frac{dh}{dt} = (u - c\sqrt{h})/A \quad (19)$$

where the state variable is the level in the tank h and the input is the volumetric flow rate entering the system u . $A = 0.23 \text{ m}^2$ denotes the cross-sectional area of the tank and $c = 0.008333 \text{ m}^{5/2}/\text{s}$ is the outlet resistance coefficient. The minimum tank height is 0 m and the maximum tank height is 0.5184 m. No disturbances were considered.

The process was initialized at $x_{init} = x(t_0) = 0.1 \text{ m}$ and numerically integrated using the explicit Euler method with an integration step of 10^{-3} s . The simulation was performed over 7 s of operation in Blender 2.93 using its embedded Python interpreter. A proportional-integral (PI) controller was designed to drive the tank level to its set-point h_{sp} over 7 s of operation. The PI controller for the tank level has the following form:

$$\frac{d\epsilon}{dt} = h_{sp} - \tilde{h}, \quad \epsilon(0) = 0 \quad (20)$$

$$u = u_s + K_c(h_{sp} - \tilde{h}) + K_c\epsilon/\tau_I \quad (21)$$

where u is the controller output ($0 \leq u \leq 0.6 \text{ m}^3/\text{s}$), $u_s = 0.0026 \text{ m}^3/\text{s}$ is the steady-state value of u that corresponds to the initial level of the tank at $t = 0$. \tilde{h} is the measured level of the tank, ϵ is the dynamic state of the PI controller. The PI tuning parameters were selected to be $K_c = 0.6$ and $\tau_I = 43.2$. The set-point was set to be $h_{sp} = 0.4 \text{ m}$.

For the image-based measurements, which are sent to the controller every sampling period ($\Delta = 0.1 \text{ s}$), a fixed camera was positioned facing one side of the tank. For the image processing task, we utilized a Python Imaging Library, Pillow [12], which includes features and supports for loading, manipulating, and saving images. Fig. 1 shows the render of the tank level at its final configuration. To measure the level of the tank, the algorithm must translate the tank image provided by the camera sensor into a measurable tank level so that the controller can compute the control actions accordingly. This can be achieved by taking the RGBA values of the pixels at the bottom and top of the tank as references for their pixel indexes to track the variations in the tank image over time. Specifically, based on the pixel indexes of the bottom and top of the tank image at $t = 0$, a linear



Fig. 1. Render of the level of the tank (converted from png to PDF to eps) at $t = 7$ s with $h_{sp} = 0.4$ m using Blender.

relation can be obtained to convert the pixel index counted at the top of the tank image to the level of the tank. The camera image of Fig. 1 has size of 1920×1080 . Variations in the pixel indexes must be counted only in the RGBA values corresponding to the tank level. The pixel index at the bottom of the tank is 1079 (which represents $h = 0$ m) and, at $t = 0$, the pixel index at the top of the liquid is 1055 (which represents $h = 0.1$ m). A linear conversion between the tank level and pixel index at the top of the tank image (I_p) was then developed and is given by Eq. 22. Therefore, at every sampling time t_k , a new count is performed to obtain the pixel index at the top of the tank image and its conversion to the measured level of the tank is sent to the PI controller as follows:

$$\tilde{h} = -3.8462 \times 10^{-3} \times I_p + 4.1577 \quad (22)$$

The change in the RGB value from the color that represents the tank level to the color outside the boundaries of the tank level is detected in the presence of lighting by allowing for several RGB values to represent the tank, and several to represent the region beyond the tank. In Blender, to update the tank level in the renders of the camera, the global coordinates where the tank is located in Blender Edit Mode must be tracked as well. In particular, the bottom edge of the tank was positioned at $(0,0,-1.57)$ m. The coordinates of the tank level at $h = 0.1$ m are $(0,0,-1.47)$ m, and thus variations in the tank level can be adjusted in Blender by updating the position of the top edge of the tank.

The closed-loop response of the tank level under this PI controller of Eqs. 20-21 described above is shown in Fig. 2. We can observe that both the actual closed-loop state and the measured tank level reach the set-point of 0.4 m after 1.5 s of operation under the PI controller.

Remark 1: The ability to capture small process changes based on image-based measurements depends on the process dynamics and the sampling period to collect the measurements. The degree to which the camera is “zoomed-in” on the process segment to be visually measured impacts the degree to which small process variations in a sampling period can be captured for measurement accuracy.

V. CONCLUSION

This work examines a control-assisted framework for model discrimination using LEMPC and a framework for testing image-based controllers using Blender. Theoretical results for the former study indicate that closed-loop stability

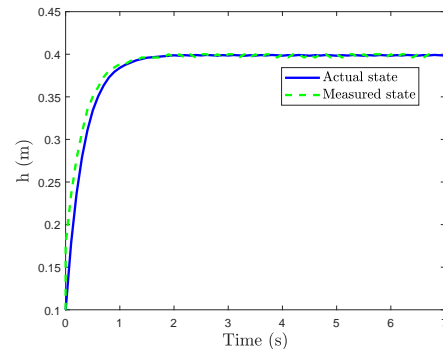


Fig. 2. Closed-loop response of the tank level problem under the image-based PI controller.

can be maintained while the models are discriminated; a level control problem was used in the second study.

ACKNOWLEDGMENT

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

REFERENCES

- [1] M. Von Stosch, R. Oliveira, J. Peres, and S. F. de Azevedo, “Hybrid semi-parametric modeling in process systems engineering: Past, present and future,” *Computers & Chemical Engineering*, vol. 60, pp. 86–101, 2014.
- [2] O. Kahrs and W. Marquardt, “The validity domain of hybrid models and its application in process optimization,” *Chemical Engineering and Processing: Process Intensification*, vol. 46, pp. 1054–1066, 2007.
- [3] S. Olofsson, E. S. Schultz, A. Mhamdi, A. Mitsos, M. P. Deisenroth, and R. Misener, “Design of dynamic experiments for black-box model discrimination,” *arXiv preprint arXiv:2102.03782*, 2021.
- [4] P. Schragl, P. Tkachenko, and L. Del Re, “Iterative model identification of nonlinear systems of unknown structure: Systematic data-based modeling utilizing design of experiments,” *IEEE Control Systems Magazine*, vol. 40, pp. 26–48, 2020.
- [5] C. Waldron, A. Pankajakshan, M. Quaglio, E. Cao, F. Galvanin, and A. Gavriilidis, “Closed-loop model-based design of experiments for kinetic model discrimination and parameter estimation: Benzoic acid esterification on a heterogeneous catalyst,” *Industrial & Engineering Chemistry Research*, vol. 58, no. 49, pp. 22 165–22 177, 2019.
- [6] H. Oyama and H. Durand, “Lyapunov-based economic model predictive control for online model discrimination,” *Computers and Chemical Engineering*, in press.
- [7] M. Heidarinejad, J. Liu, and P. D. Christofides, “Economic model predictive control of nonlinear process systems using Lyapunov techniques,” *AIChE Journal*, vol. 58, pp. 855–870, 2012.
- [8] T. Pearson, D. Brabec, and S. Haley, “Color image based sorter for separating red and white wheat,” *Sensing and Instrumentation for Food Quality and Safety*, vol. 2, pp. 280–288, 2008.
- [9] B. Lin, B. Recke, J. K. Knudsen, and S. B. Jørgensen, “Bubble size estimation for flotation processes,” *Minerals Engineering*, vol. 21, pp. 539–548, 2008.
- [10] L. Giuliani and H. Durand, “Data-based nonlinear model identification in economic model predictive control,” *Smart and Sustainable Manufacturing Systems*, vol. 2, pp. 61–109, 2018.
- [11] *Blender*, Blender Foundation, Stichting Blender Foundation, Amsterdam, 2021. [Online]. Available: <http://www.blender.org>
- [12] A. Clark, “Pillow (PIL fork) documentation,” 2015. [Online]. Available: <https://buildmedia.readthedocs.org/media/pdf/pillow/latest/pillow.pdf>