Abstract—This paper investigates the data-driven predictive control problems for a class of continuous-time industrial processes with completely unknown dynamics. The proposed approach employs the data-driven technique to get the system matrices online, using input-output measurements. Then, a model-free predictive control approach is designed to implement the receding-horizon optimization and realize the reference tracking. Feasibility of the proposed algorithm and stability of the closed-loop control systems are analyzed, respectively. Finally, a simulation example is provided to demonstrate the effectiveness of the proposed approach.

Index Terms—Data-driven control, industrial processes, model predictive control (MPC), reference tracking.

I. INTRODUCTION

As a practically effective approach, model predictive control (MPC), or receding horizon control, has attracted notable attention in the field of industrial process control [1]. To deal with optimal control problems, MPC can allow for industrial processes uncertainties and constraints much more straightforwardly than other methods [2], [3]. The core of all model-based predictive algorithms is to use “open-loop optimal control” instead of “closed-loop optimal control” within a moving horizon [4]. It brings a lot of robustness and reliability to allow the controller have the ability to recognize the control process. But, the acquisition of knowledge of an a priori model seriously affects those performances of MPC. The practically measured data comes from the complicated processes and its utilization into MPC will greatly facilitate the design procedures, avoiding the need for initially accurate dynamic models [5].

Nowadays, many research works focus on the data-driven predictive control, that is applicable on-line both to regulation and tracking control problems [6]–[17]. More earlier works can refer to [18] and [13]. A Markov data-based LQG control algorithm is suggested in [18] and data-based optimal control based on the system’s Markov parameters is provided in [13]. Both of them utilize the prior measurements to design the predictive control, along with its implementation on-line. Many results (see, [9]–[11]) still need some knowledge of the system information to fit the structure of the process model. Other works, like [19], focus on the polytopic uncertain systems with unmeasurable system states and a convex hull needs to be known. In [16], [20], data-driven subspace approach is introduced to design the predictive controller and in [8], [21], reinforcement learning approach is used to reduce the model-based dependence on predictive controller design procedures. We remark that all the aforementioned predictive control methods are designed based on the accurate model, as well as adequate uncertainty description of the linear or non-linear plant of the processes.

In this paper, the limitations of them are circumvented. We use the adequately measured data from the complicated industrial processes following the methods presented in [22], [23] and [24], [25]. To be precise, the data-driven learning technique of [22] will be employed to iteratively approximate the dynamical parameters, without requiring the a priori knowledge of the system matrices. Then, the linear plant’s version of [26] will be applied to predict the future trajectories, by following the continuous-time predictive control approach of [7] but removing the assumption on partial knowledge of the system dynamics. Under this framework, the data-driven predictive control input can be generated on-line and can be used for the control of time-varying or nonlinear plants, since the algorithm is able to adapt to the actual dynamics by obtaining a linear model of the system at each sample. The contributions of the paper are three-fold. (i) A data-driven approach is designed to adaptively approximate the system matrices, without requiring the a priori knowledge of the system. (ii) A continuous-time data-driven MPC approach is developed for the continuous-time linear system, using repeatedly the state and input formation on some fixed time intervals. (iii) By implementing the proposed data-driven predictive control algorithm, both recursive feasibility of the optimization problem and closed-loop stability of the whole system are guaranteed.

The rest of this paper is organized as follows. In Section II, the problems are briefly formulated. In Section III, the data-driven predictive control approach with completely unknown dynamics algorithm is presented, and the feasibility and stability analysis are conducted. In Section IV, we apply the proposed approach to the optimal control problem of two continuous stirred tank reactor (CSTR). Conclusions are given in Section V.

Notation: Through this note, \( \mathbb{R} \) denotes the set of real numbers, \( | \cdot | \) represents the Euclidean norm for a vector and the induced norm for a matrix. For a \( a \in \mathbb{R}, R_{\geq a} \)}
denotes the interval \([a, \infty)\) and \(\mathbb{R}_{\geq 0}\) denotes the interval \((a, \infty)\). For any vector \(x\), \(x^T\) denotes its transpose and \(\|x\|_p^p\) represents \(x^TPx\) for a real symmetric and positive defined matrix \(P\). For a real symmetric matrix \(A \in \mathbb{R}^{n \times n}\), \(A > 0\) or \(A \succeq 0\) means that \(A\) is positive definite or semi-positive definite, \(\lambda_M(A)\) and \(\lambda_m(A)\) denote the maximum and minimum eigenvalue of \(A\), respectively. \(\otimes\) indicates the Kronecker product operator, \(\text{vec}(\cdot)\) denotes the vectorization operator and \(\text{vec}^{-1}(\cdot)\) denotes the converse vectorization operator, i.e., \(\text{vec}(A) = [a_1^T, \ldots, a_m^T]^T\) and \(\text{vec}^{-1}[a_1^T, \ldots, a_m^T] = A\), where \(a_i \in \mathbb{R}^n\) are the columns of a matrix \(A \in \mathbb{R}^{n \times m}\). A continuous function \(\alpha: \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}\) is said to be a \(K\) function if it is strictly increasing, and \(\alpha(s) > 0\) for \(s > 0\) with \(\alpha(0) = 0\). A continuous function \(\alpha(\cdot)\) is said to be a \(K_\infty\) function if it is a \(K\) function, and \(\alpha(s) \rightarrow \infty\) for \(s \rightarrow 0\). For any piecewise continuous function \(u(\cdot): \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^m\), \(u^k\) denotes the \(r\)th order derivatives of \(u(\cdot)\).

### II. Problem formulation and preliminaries

Consider a continuous-time industrial process control system with the form

\[
\begin{align*}
\dot{x}(t) &= Ax(t) + Bu(t), \quad x(t_0) = x_0, \\
y(t) &= Cx(t),
\end{align*}
\]

where \(t \geq t_0, x \in \mathbb{R}^n\) is the measurable state; \(u \in \mathcal{U} \subset \mathbb{R}^m\) is the control input; \(y \in \mathbb{R}^q\) is the controlled output fully available for feedback control design. Assume that \(\mathcal{U} \subset \mathbb{R}^m\) is a nonempty compact convex set and contains the origin as its interior point. \(A \in \mathbb{R}^{n \times n}\) and \(B \in \mathbb{R}^{n \times m}\) are unknown system matrices with \((A, B)\) controllable, \((A, C)\) observable, satisfying \(|A| \leq A_M, |B| \leq B_M\).

Some standard assumptions are made on \((1)\) and \((2)\). Similar assumptions can be found in \([7, 22, 26]\) for solving (cooperative) output tracking problems.

**Assumption 1.** The input constraint has the box constrain form as \(\underline{u} \leq u \leq \bar{u}\) with elementwise inequality and \(\underline{u}, \bar{u}\) the respective lower and upper bounds.

**Assumption 2.** There exists a constant matrix \(K_0\) such that \(A - BK_0\) is a Hurwitz matrix with \(-K_0x(t) \in \mathcal{U}\).

**Assumption 3.** The input relative degree (IRD) of system \((1)\) is defined as \(\rho\).

**Remark 1.** In Assumption \((1)\) we refer the set \(\mathcal{U}\) as a box constrain, which accurately describes nearly any set of standard mechanical actuators. Assumption \((2)\) is made such that the initially feasibility can be achieved for the system \((1)\). Assumption \((3)\) imposes IRD for the system \((1)\), which is used for simplify the solving of regulation problem \((7)\).

For \((1)\) and \((2)\), the output of the system should track the given reference \(y_d(t), t \geq 0\). The tracking error can be given as \(e(t) = y(t) - y_d(t), t \geq 0\). To that end, a sampled-data MPC, which is based on the repeated solution of an open-loop optimal control problem, is provided in this paper. At each time instant \(t = t_k\), the state \(\hat{y}(t_k)\) is measured and then, the controller predicts the system behavior in the future over a prediction horizon \(T\) by minimizing a certain objective cost function. The procedure is repeated at every sampling time instant \(t_k\) for \(k = 1, 2, \ldots\)

For the system \((1)\) and \((2)\), the cost function \(J(t_k) := J(x(t_k), y_d(t), \hat{u}_k(t))\), at time \(t_k\), is defined as

\[
J(x(t_k), y_d(s), \hat{u}_k(s)) = \int_{t_k}^{t_k+T} L(x(t_k), y_d(s), \hat{u}_k(s))ds + F(y_d(t_k + T), \hat{y}(t_k + T))
\]

where, \(L(\cdot, \cdot, \cdot)\) and \(F(\cdot, \cdot)\) denote the stage and terminal cost functions with the from

\[
L(x(t_k), y_d(t), \hat{u}_k(t)) = \|e(t)||_Q^2 + \|\hat{u}_k(t)||_R^2, \quad t \geq 0,
\]

where \(Q = Q^T > 0\) and \(R = R^T \succeq 0\) are symmetric and sign definite weight matrices.

Then, at time \(t_k\), the optimal control signal \(\hat{u}_k^*(s), s \in [t_k, t_k + T]\), is obtained by solving the following finite-horizon optimal control problem as

\[
\begin{align*}
\min_{\hat{u}_k(s) \in \mathcal{U}} & \quad J(x(t_k), y_d(s), \hat{u}_k(s)) \\
\text{s.t.} & \quad \dot{\hat{x}}(s) = \mathcal{H}(\hat{x}(s), \hat{u}_k(s)) \Theta \\
& \quad \hat{y}(s) = C\hat{x}(s), \hat{x}(t_k) = x(t_k) \\
& \quad \hat{u}(s) \in \mathcal{U}, \quad s \in [t_k, t_k + T]
\end{align*}
\]

where \(\mathcal{H}(\cdot, \cdot): \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^{n(n+m)}\) is defined as

\[
\mathcal{H}(x, u) = [(x \otimes I_n)^T (u \otimes I_n)^T]
\]

and \(\Theta\) denotes the vector of the dynamical parameters for \((1)\), defined as

\[
\Theta = [\text{vec}(A)^T \text{vec}(B)^T]^T \in \mathbb{R}^{n^2 + nm}
\]

where \(\text{vec}(\cdot)\) denotes the vectorization operator, defined as for any \(P = [p_{ij}] \in \mathbb{R}^{n \times l}, l \in \{n, m\}\), \(\text{vec}(P) = P\), where

\[
\hat{P} = [p_{11}, p_{21}, \ldots, p_{n1}, p_{12}, \ldots, p_{n2}, \ldots, p_{nl}]^T
\]

If the controller has the exact knowledge of system matrices \(A\) and \(B\), then we can directly solve the finite-time optimal control problem \((4)\) by \((22)\) to get the desired predictive control policy \(\hat{u}_k^*\). But, in this paper, we seek a data-driven approach to remove the model-based dependence of \((6)\) on the predictive controller design. Thus, the primary objective of this paper is to explore an online learning-based approach to find the data-driven predictive control policy, without requiring any knowledge of the system dynamics, just by using \(a\) priori data of inputs and outputs.

### III. Data-driven Predictive Control with Completely Unknown Dynamics

In this section, to facilitate the predictive controller design, the state \(\hat{x}\) and the system parameters \(\Theta\) of the predicted model \((4a)\) are both estimated from the input-output measurements, using a simultaneous state and parameter estimator. Our proposed online-learning strategy does not rely on \(\Theta\), i.e., either \(A\) or \(B\), which is totally data-driven approach.
A. Dynamical Parameters Estimator

To obtain the dynamical parameters $\Theta$ in (6), i.e., $A$ and $B$, we present our online learning strategy with the previous input-output data. By using more prior data for continuous-time system (1) than the instants of solving (4), we try to seek a least-squares optimization solution to get the estimations of all the dynamical parameters $\Theta$. To that end, each time instant $t = t_k$, with enough prior data collected, we can find a minimal periodic time interval $\delta t$, such that for some $t_j < t_k$ of $j \in \{1, 2, ..., k - 1\}$, there exists a constant integer $N_k$ satisfying $N_k \delta t = t_k - t_j$.

To begin with, for any time instant $t \leq t_k$, doing the integrals at the time interval $[t - \delta t, t]$ along the trajectory of (1) leads to

$$x(t) - x(t - \delta t) = A \int_{t-\delta t}^{t} x(t) \, dt + B \int_{t-\delta t}^{t} u(t) \, dt$$

$$= \mathcal{H} \left( \int_{t-\delta t}^{t} x(t) \, dt, \int_{t-\delta t}^{t} u(t) \, dt \right) \Theta$$

By rearranging (8), we have the linear error system in the form of

$$\mathcal{F}(t) = \mathcal{G}(t) \Theta, \quad \forall t \in [t_i, t_k]$$

where the matrices $\mathcal{F}(\cdot) : \mathbb{R} \rightarrow \mathbb{R}^n$ and $\mathcal{G}(\cdot) : \mathbb{R} \rightarrow \mathbb{R}^{n \times n}$ are defined,

$$\mathcal{F}(t) = \begin{cases} x(t) - x(t - \delta t), & t \in [\delta t, \infty) \\ 0, & t < \delta t \end{cases}$$

$$\mathcal{G}(t) = \left( \mathcal{A}(t) \otimes I_{n} \right)^T \left( \mathcal{B}(t) \otimes I_{n} \right)^T$$

where $\otimes$ denotes the Kronecker product, and the vectors $\mathcal{A}(\cdot) : \mathbb{R} \rightarrow \mathbb{R}^n$ and $\mathcal{B}(\cdot) : \mathbb{R} \rightarrow \mathbb{R}^{n \times n}$ are defined,

$$\mathcal{A}(t) = \begin{cases} \int_{t-\delta t}^{t} x(\tau) \, d\tau, & t \in [\delta t, \infty) \\ 0, & t < \delta t \end{cases}$$

$$\mathcal{B}(t) = \begin{cases} \int_{t-\delta t}^{t} u(\tau) \, d\tau, & t \in [\delta t, \infty) \\ 0, & t < \delta t \end{cases}$$

By (12)-(13), we note that for any time instant $t \leq t_k$, by using the previous input-output measurements $x(\tau)$ and $u(\tau)$, $\tau \leq t$, $\mathcal{F}(t)$ and $\mathcal{G}(t)$ in (10)-(11) are both available. Then, by taking periodic sampling $\delta t$ for (1) at each time $t = t_k - i \delta t$ with $i \in \{0, 1, 2, ..., N_k\}$, such that $t_j \leq t_k$, then we can get a set of composite data,

$$\mathcal{D}_k = \bigcup_{i} \{ \mathcal{F}_i, \mathcal{G}_i \}_{i=1}^{N_k}$$

which satisfies

$$\mathcal{F}_i = \mathcal{G}_i \hat{\Theta} + \epsilon, \quad \forall i \in \{1, 2, ..., N_k\}$$

where $\mathcal{F}_i = \mathcal{F}(t_k - i \delta t)$, $\mathcal{G}_i = \mathcal{G}(t_k - i \delta t)$, $\hat{\Theta} = [\hat{A}^T \hat{B}^T]^T$ is an estimate of $\Theta$ in (6) and $\epsilon$ is the estimation error due to the data-driven approximation of $\Theta$ by using $\mathcal{D}_k$.

To bring $\epsilon$ to its minimum value, we have the following optimization problem (OP):

$$\min_{\hat{\Theta}} \epsilon^T \epsilon$$

subject to (15) and (9) - (13)

Then, an adaptive least-square method is used here to give a solution to problem (16). But, before that, the full rank of dataset $\mathcal{D}_k$ is defined as follows. To guarantee the existence of the solution of (16), we also give the full rank condition for dataset $\mathcal{D}_k$.

**Definition 1.** At each time instant $t_k$ for some integer $N_k$, the data stack $\mathcal{D}_k$ is said to have full rank, if there exists an integer $N_0 > 0$, such that for all $N_k \geq N_0$, we have the matrix $\mathcal{A}_k$, defined as $\mathcal{A}_k := \sum_{i=1}^{N_k} \mathcal{G}_i^T \mathcal{G}_i \in \mathbb{R}^{(n^2 + nm) \times (n^2 + nm)}$, satisfying

$$\text{rank}(\mathcal{A}_k) = n^2 + nm$$

Then, we have the following lemma to give a sufficient condition to guarantee the full rank property of dataset $\mathcal{D}_k$. The proof is similar to [28], and here it is omitted.

**Lemma 1.** A data stack $\mathcal{D}_k$ has full rank, if there exists a constant $d > 0$, such that

$$0 < d < \gamma_m(\mathcal{A}_k)$$

Assume that the measurements of the inputs and the outputs are prior collected at an enough large number $N_k \gg n^2 + nm$ points of time $t_k - i \delta t$ with $i \in \{0, 1, ..., N_k\}$, which makes data stack $\mathcal{D}_k$ has full rank by Lemma III-A. Then, at time $t = t_k$, one can use the data stack $\mathcal{D}_k$ to evaluate the unknown dynamical parameters $\Theta$ in (6). By solving (16), the following learning-based update law is obtained

$$\dot{\hat{\Theta}} = \eta_\theta \sum_{i=1}^{N_k} \mathcal{G}_i^T (\mathcal{F}_i - \mathcal{G}_i \hat{\Theta})$$

where $0 < \eta_\theta \in \mathbb{R}$ is a constant learning rate.

For ease of exposition, the original continuous-time system (1) is expressed in the form of,

$$\dot{x}(t) = \mathcal{H}(x(t), u(t)) \Theta$$

By considering the linear error system with parameters update law (19), it follows that

$$\dot{x}(t) = \mathcal{H}(x(t), u(t)) \hat{\Theta} + w(x(t), u(t))$$

$$:= \tilde{A} x(t) + \tilde{B} u(t) + w(t)$$

where $\tilde{B} = \text{vec}^{-1}(\tilde{B})$, $\tilde{A} = \text{vec}^{-1}(\tilde{A})$, $\text{vec}^{-1}()$ denotes the converse vectorization operator, that is, for any vector $\tilde{P} \in \mathbb{R}^{n \times l}$ in (7), we have $\text{vec}^{-1}(\tilde{P}) = \tilde{P} = [p_{ij}] \in \mathbb{R}^{n \times l}$. And $w(t) := w(x(t), u(t))$ is the continuous approximation error resulting from parameters uncertainty (19). If $\epsilon = 0$, then $\hat{\Theta} = \Theta$, it implies $\tilde{A} = A$ and $\tilde{B} = B$, thus we have $w(x, u) = 0$.

The following theorem analyses the property of the term $w$ with respect to (21).

**Theorem 1.** The approximate error $w(t)$ in (20) is slowly time-varying, bounded and satisfies $\lim_{t \rightarrow \infty} w(t) = 0$.

**Proof.** By considering the closed-loop dynamics (20) and (21), we can refer to $w(t)$ as the unknown disturbance caused by the parameters uncertainty of (19). Then, $w(t)$ satisfies

$$w(t) = \mathcal{H}(x(t), u(t)) (\hat{\Theta} - \Theta)$$

Letting $\tilde{\Theta} = \hat{\Theta} - \Theta$ and bringing (19) to (22) lead to

$$|w(t)| \leq |\mathcal{H}(x(t), u(t))||\tilde{\Theta}|$$
Remark 2. Note that in (21), the conventional receding horizon expression $Ax(t) + Bx(t)$ depending on the unknown matrices $A, B$ is replaced by the term $H(x(s), u_k(s))$, where $\Theta$ can be obtained by repeatedly learning from the states and inputs measurements. Furthermore, this learned results will not affects the convergence of the system by Lemma III-A. Therefore, (19) plays an important role in identifying the system dynamics from the a prior data. As a result, the requirement of the system matrices in predicting the behavior of (11) can be replaced by the state and input information measured online.

B. Receding-horizon Optimization

To facilitate the data-driven predictive controller design for the system (21) with the dynamical parameters estimator (19), the receding-horizon predictive control problem of (4), at time instant $t = t_k$, can be reformulated,

$$\hat{u}_k(s) = \arg \min_{\hat{u}(t) \in U} J(x(t_k), y_d(s), \hat{u}_k(s))$$

subject to

$$\hat{x}(s) = \hat{A}s + \hat{B}\hat{u}_k(s), \quad s \in [t_k, t_k + T].$$

(25a)

$$\hat{y}(s) = C\hat{x}(s), \hat{x}(t_k) = x(t_k), \quad \hat{u}_k(s) \in U, \quad s \in [t_k, t_k + T].$$

(25b)

To solve the optimization problem (25), under Assumption [11], we define the decision variables as $\hat{u}_k(s) = [\hat{u}_k^T(s), (\hat{u}_k^1)^T(s), \ldots, (\hat{u}_k^r)^T(s)]$ for some control order $r$ larger than $\rho \geq 1$. Note that the first term of $\hat{u}_k(\tau)$ is the to-be-optimized control input $\hat{u}_k(\tau)$ in (25). More generally, for the control law $\hat{u}_k(\tau)$ with a large enough control order $r$, we let $\hat{u}_k^l(\tau) = 0$ for any integer $l \geq r$.

Then, for the output prediction of optimization problem (25), by following (2), the future output $y_d(s) = y(t + \tau), t = t_k, k = 1, 2, \ldots$, in the moving horizon $\tau \in [0, T]$ is approximated by Taylor series expansion,

$$y(t + \tau) = y(t) + \tau y^1(t) + \cdots + \frac{\tau^r}{r!} y^r(t) + O(\tau^r)$$

where the $i$-th derivative of the output $y^i(t)$ with $i \in \{1, 2, \ldots, \rho, \ldots, r\}$ is obtained by

$$y^i = C\hat{A}^i x + \sum_{k=0}^{i-1} C\hat{A}^{i-k} W_k^i, i = 1, \ldots, \rho - 1$$

(27)

By rewriting the output $y(t + \tau)$ in a compact form, it follows that

$$y(t + \tau) = [T_1(\tau) \quad T_2(\tau)] [Y_1] [Y_2]$$

(29)

where

$$T_1(\tau) = \begin{bmatrix} 1, \tau, \ldots, \frac{\tau^{\rho-1}}{(\rho-1)!} \end{bmatrix}, \quad T_2(\tau) = \begin{bmatrix} y^1(T), y^2(T), \ldots, y^r(T) \end{bmatrix}^T.$$ 

(30)

and $A_1, A_2, A_3$ are defined,

$$A_1 = \begin{bmatrix} C \hat{A} & C \hat{A} \hat{B} & \cdots & \hat{B} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{C} A^r & \hat{C} A^{r-1} & \cdots & \hat{C} A \end{bmatrix}, \quad A_2 = \begin{bmatrix} C \hat{A} & \cdots & \hat{B} \\ \vdots & \ddots & \vdots \\ \hat{C} A^r & \cdots & \hat{C} A \end{bmatrix},$$

(32)

$$A_3 = \begin{bmatrix} C \hat{A} \hat{B} & \cdots & \hat{B} \\ \vdots & \ddots & \vdots \\ \hat{C} A^r \hat{B} & \cdots & \hat{C} A \end{bmatrix}.$$ 

(33)

Then, for the reference signal $y_d(t + \tau)$, by Theorem 1, we have $w^k(t + \tau) \approx 0$ for $k = 1, 2, \ldots, r$. Thus, $y_d(t + \tau)$ satisfies (29)-(31) with $Y_1 = [y^1(T), y^2(T), \ldots, y^r(T)]^T$ and

$$Y_2 = [(y^{[1]}(T), y^{[2]}(T), \ldots, y^{[r]}(T)]^T.$$ 

(34)

By (34)-(35) into the given performance index (3), one can go to,

$$J(t_k) = \int_0^T \left[ \hat{Y}_1^T \hat{Y}_2 \right] \left[ \xi_1(\tau) \xi_2(\tau) \right] \left[ \dot{Y}_1 \dot{Y}_2 \right] d\tau + \int_0^T \hat{u}^T \dot{u} d\tau + F(\hat{Y}_1(t_k + T))$$

(36)
where $\hat{Y}_i = Y_i - Y_{i,d}$ for $i \in \{1, 2\}$, and $\Xi_0(\tau) = \sqrt{\mathcal{Q} T_\tau(\tau)}$ for $i \in \{1, 2\}$. By defining $\mathcal{T}_{i,j} = \int_0^T \Xi_j^T \Xi_j d\tau$ with $i, j \in \{1, 2\}$ and $\mathcal{T} = \int_0^T T^R T d\tau$, we note that $\mathcal{T}_{1,2} = \mathcal{T}_{2,1}$. Thus, the performance index (37) can also be rewritten as

$$J(t_k) = \hat{Y}_1^T \mathcal{T}_{1,2} \hat{Y}_1 + \hat{Y}_2^T \mathcal{T}_{2,1} \hat{Y}_2$$

$$+ u^T F(\hat{Y}_1(t_k) + T))$$

Due to equation (34), taking partial derivative of $J(t_k)$ with respect to $\tilde{u}_k$ yields

$$\frac{\partial J}{\partial \tilde{u}} = 2 \left( \left( \frac{\partial \hat{Y}_2}{\partial \tilde{u}} \right) \mathcal{T}_{2,2} \hat{Y}_2 + \mathcal{T}_4 \right) \tilde{u} + 2 \left( \left( \frac{\partial \hat{Y}_1}{\partial \tilde{u}} \right) \mathcal{T}_{1,2} \hat{Y}_1 \right)$$

$$+ 2 \mathcal{B}_1^T \mathcal{T}_{2,2} \left( A_2 x - Y_{2,d} \right)$$

By letting $\frac{\partial J}{\partial \tilde{u}} = 0$, we can get the optimized predictive control law $\tilde{u}_k^*$ as

$$\tilde{u}_k^* = - \left( \mathcal{B}_1^T \mathcal{T}_{2,2} \mathcal{B}_3 + \mathcal{T}_4 \right)^{-1} \mathcal{B}_1^T \left( \mathcal{T}_{2,2} \hat{Y}_2 - \mathcal{T}_{1,2} \hat{Y}_1 \right)$$

where $\hat{Y}_2 = Y_{2,d} - A_2 x$. Then, taking the first row of the optimized control law (39), the continuous-time predictive control law, applied to the plant, is given by

$$\tilde{u}_k^*(t) = I_u \tilde{u}_k^*$$

where $I_u = [1, 0, ..., 0]_1 \times (r+1)$.

**Remark 3.** Note that from (40), the existence of the optimized solution $\tilde{u}_k^*$ depends on the reversibility of matrix $\mathcal{M} = \mathcal{B}_1^T \mathcal{T}_{2,2} \mathcal{B}_3 + \mathcal{T}_4$, with $\mathcal{B}_3$ computed from (32), $\mathcal{A}$ and $\mathcal{B}$ calculated from (19). Thus, before we implement the receding-horizon optimization, we first check the reversibility of matrix $\mathcal{M}$ by removing the repeated columns of dataset $\mathcal{D}_k$, only left the distinct columns.

**C. Handling constraints**

To deal with optimal control problems, MPC can allow for industrial processes uncertainties and constraints much more straightforwardly than other methods [2], [3]. Assumption [11] will allow us to use a specialized active-set algorithm which is more efficient and easier to implement. A box-constraint solver can be immediately generalized to any linear inequality constraints using slack variables. In the following, we formalize two classical ways to enforce the control limits.

1) **Saturating Functions:** A conventional attempt to enforce box constraints is to clamp the controls in the forward-pass. The element-wise clamping, or projection operator, is denoted by $sat()$, which is the input saturation function defined as

$$sat(u) = \begin{cases} sat(u_1) \cdot sat(u_2) \cdot \cdots \cdot sat(u_m) & \text{if } sat(u) \neq \text{ either } 1 \text{ or } -1 \\ sat(u_1) \cdot sat(u_2) \cdot \cdots \cdot sat(u_m) & \text{if } sat(u) = \text{ either } 1 \text{ or } -1 \\ \end{cases}$$

$$sat(u_i(t)) = \begin{cases} u_i(t) & \text{if } u_{i,min} < u_i(t) < u_{i,max} \\ u_{i,min} & \text{if } u_i(t) \leq u_{i,min} \\ u_{i,max} & \text{if } u_i(t) \geq u_{i,max} \\ \end{cases}$$

with $u = [u_1 \ u_2 \ \cdots \ u_m]$, and $u_{i,min} \leq 0$ and $u_{i,max} \geq 0$ are the boundaries of $i$th control input of system (1).

It implies that it is tempting to simply replace the obtained control in the forward-pass with

$$\tilde{u} = sat(u^*)$$

However, the corresponding search direction may not be a descent direction anymore, harming convergence.

2) **Squashing Functions:** Another way to enforce box constraints is to introduce a sigmoidal squashing function $s(u)$ on the controls

$$x_{i+1} = f(x_i, s(u_i))$$

where $s(\cdot)$ is an element-wise sigmoid with the vector limits

$$\lim_{u \to -\infty} s(u) = u, \quad \lim_{u \to -\infty} s(u) = \pi$$

For example, $s(u) = \frac{e^{u}-e^{-u}}{e^{u}+e^{-u}}$ is such a function. A cost term should be kept on the original $u$ and not only on the squashed $s(u)$. Otherwise it will reach very high or low values and get stuck on the plateau. An intuition for the poor practical performance of squashing is given by the nonlinearity of the sigmoid. Since the backward pass uses a locally quadratic approximation of the dynamics, significant higher order terms will always have a detrimental effect on convergence.

**D. Our Proposed Algorithm**

Our proposed method is summarized as the following Algorithm 1.

**Algorithm 1:** Data-driven predictive control algorithm

**Data:** current period $t$; a initially stable control

$u^* = -K_0 x$; prediction horizon $T$; terminal cost $F$;

**Result:** Optimal MPC input $u_i^*$

1. $t_k = k \leftarrow 0$;

2. Collect data $\mathcal{D}_k$ in (14);

3. while do

4. Collect the data and form $\mathcal{D}_k$;

5. if (18) is satisfied then

6. Generate the estimator (19) by using $\mathcal{D}_k$;

7. Implement (25) to obtain the optimized control (40);

8. Time evolves continuously with $t$;

9. end

10. $k \leftarrow k + 1$;

11. end

**E. Performance Analysis**

Before proceeding further, we first introduce the following definition and lemma.

**Definition 2.** For the system (21), given a compact set $\mathcal{E}$, with $\{0\} \subset \mathcal{E} \subset \mathbb{R}^n$ and $\mathcal{E}$ being a robustly positively invariant set, if there exists a positive definite function $V(\cdot) : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$, such that,

$$V(x) \geq \alpha_3(|x|), \quad V(x) \leq \alpha_2(|x|) + c_1$$

$$V(\dot{x}) - V(x) \leq -\alpha_3(|x|) + \alpha_4(|x|) + c_2$$

(42)  \quad (43)
for all \( t \in \mathbb{R}_{\geq 0} \) with \( \alpha_1, \alpha_2, \alpha_3 \) being \( K_{\infty} \) function, \( \alpha_4 \) being \( K \) function, and \( c_1, c_2 \geq 0 \). Then, the function \( V(\cdot) \) is a regional input-to-state practical stability (ISpS)-type Lyapunov function in \( \mathcal{E} \) for the system.

Based on Definition 2, we can have the following lemma, directly borrowed from [7].

**Lemma 2.** Given a robust positively invariant set \( \mathcal{E} \) for the system (21), if it admits an ISpS-type Lyapunov function \( V(\cdot) \), then the system is regional ISpS in \( \mathcal{E} \), and all the signals of the closed-loop system with the control input \( u_E \) are bounded, where \( u_E(t) \) denotes the control such that the set \( \mathcal{E} \) is an invariant region satisfying the constraints.

Note that by Assumption 1, the developed optimization problem of data-driven predictive control in (25) is initially feasible with \( u_0 = -K_0x \), then the global stability can be proved by using the Lemma 2.

**Theorem 2.** Suppose the Assumption 1 and 2 hold for system (21). Consider any time \( t_k \) such that the problem of (25) has a solution and the optimal input \( \hat{u}_k \) is implemented for time \([t_k, t_{k+1})\). Assumed that at \( t_{k+1} \), \( \hat{y}(t_{k+1}) = y(t_{k+1}) \). Therefore, the remaining piece of optimal input \( \hat{u}_k(s), \tau \in [t_{k+1}, t_{k+1} + T] \) satisfies the input constraints. Thus, we construct the control input as,

\[
\hat{u}_{k+1}(\tau) = \begin{cases} 
\hat{u}_k(\tau), & \tau \in [t_{k+1}, t_{k+1} + T] \\
u_E(\tau), & \tau \in [t_k + T, t_{k+1} + T]
\end{cases}
\]

where \( u_E(\tau) \) makes the desired reference reached and the constraints satisfied. Thus, the predictive control problem is feasible at \( t_{k+1} \). It implies that feasibility of the problem at \( t_k \) implies the recursive feasibility at \( t_{k+1} \).

**Convergence:** Let the optimal cost function at \( t_k \) as the value function \( V(x(t_k)) = \mathcal{J}(x(t_k), y_{d}(s), \hat{u}_k(s)) \). If \( V(x(t_k)) \) is strictly decreasing, the tracking error \( e \) will converge to the origin. To this end, we write the value function at \( t_k \) as,

\[
V(t_k) = \int_{t_k}^{t_{k+1}} (\|e(\tau)\|_Q^2 + \|\hat{u}_k(\tau)\|_R^2) \, d\tau + F(y_d(t_k + T), \hat{y}(t_k + T))
\]

Then, by applying \( \hat{u}_{k+1}(t) \) in (44) to the system, beginning from \( y(t_{k+1}) \), one has

\[
J(t_{k+1}) = \int_{t_k}^{t_{k+1} + T} (\|e(\tau)\|_Q^2 + \|\hat{u}_{k+1}(\tau)\|_R^2) \, d\tau + F(y_d(t_{k+1} + T), \hat{y}(t_{k+1} + T))
\]

By substituting (45) in (46), it follows,

\[
J(t_{k+1}) = V(t_k) - \int_{t_k}^{t_{k+1}} (\|e(\tau)\|_Q^2 + \|\hat{u}_k(\tau)\|_R^2) \, d\tau - F(y_d(t_k + T), \hat{y}(t_k + T))
\]

Further, based on \( V(t_{k+1}) = J(t_{k+1}) \), we obtain,

\[
V(t_k) \leq V(t_{k+1}) \leq -\int_{t_k}^{t_{k+1}} (\|e(\tau)\|_Q^2 + \|\hat{u}_k(\tau)\|_R^2) \, d\tau
\]

It implies that \( V(t_{k+1}) \) is strictly decreasing. Hence, the proof is complete.

IV. APPLICATION TO TWO-CSTR PROCESS

Consider two continuous stirred tank reactor (CSTR) system with a full description in [23, 30]. The open-loop model is a six-state continuous model. The system matrices \( A \) and \( B \) are directly taken from [23, 30], described in the form of (1), as you can see in [50]. The system output variables \( y_1 = 362.995x_2 \) and \( y_2 = 362.995x_4 \), denoting the two tank outlet temperatures. The control problem is to maintain the two tank temperatures at desired values \( y_d(t) = [y_{d1}(t) y_{d2}(t)]^T \), where \( y_{d1}(t) = 10 \) when \( 0 \leq t < 5s \) and \( y_{d1}(t) = 7 \) when \( t \geq 5s \), \( y_{d2}(t) = 10 \) when \( 0 \leq t < 5s \) and \( y_{d2}(t) = 4 \) when \( t \geq 5s \). The constraints is,

\[
U = \{u = [u_1 u_2]^T : |u_1| \leq 80, \quad |u_2| \leq 70\}
\]

In order to illustrate the efficiency of the proposed approach, the precise knowledge of \( A \) and \( B \) is not used in the design of the predictive controllers. Since the physical system is not stable, the initial stabilizing feedback gain is set as \( K_0 \),

\[
K_0 = \begin{bmatrix}
-4.8949 & -3426.8 & -158.1712 & -0.0320 & -43.7963 & -1.4675 \\
0 & 0 & 8.2934 & 1.1730 & 2.3886 & 104.8756
\end{bmatrix}
\]

The weighting matrices \( Q \) and \( R \) are set to be \( Q = diag([10 10 10 10 10 10]) \) and \( R = diag([1 1]) \), respectively. In the simulation, the initial values are selected at the origin. The state and input information is collected over each interval of 0.01s. When time arrives at \( t = 2s \), all the inputs and outputs are repeatedly used to approximate the matrices \( A \) and \( B \) with \( \eta_0 = 0.85 \). The predictive control also starts at \( t = 2s \) with the prediction horizon \( T = 1s \). Since then, the control input is immediately updated by solving the problem (25), and the convergence of of \( A_k := \hat{A} \) and \( B_k := \hat{B} \) to their actual values is attained after 10 iterations. The procedure of solving (25) is repeated over a fixed interval of 0.1s. The convergence of \( A_k \) and \( B_k \) to their actual values is illustrated in Fig.1. The trajectories of the output variables and the flow
rates are shown in Fig. 2. It can be seen that the data-driven predictive control algorithm can stabilize the system, without requiring the system matrices.

V. Conclusion

In this paper, a data-driven predictive control approach for continuous-time linear system with completely unknown dynamics has been provided. This method solves the infinite-horizon optimal control problem, using the system inputs and outputs information collected online, without knowing the system matrices. The methodology developed in this paper may serve as a computational tool to study the finite-horizon adaptive optimal control of uncertain nonlinear systems. Some related work has appeared in [15], which was developed using neural networks, and also in our recent work [26], which proposed a framework of distributed MPC to handle the asynchronous communication, using the a priori information associated with the interconnected neighbors to a distributed optimal design.

REFERENCES

[1] P. Karamanakos, T. Geyer, N. Oikonomou, F. D. Kieferndorf, and S. Manias, “Direct model predictive control: A review of strategies that achieve long prediction intervals for power electronics,” IEEE Industrial Electronics Magazine, vol. 8, no. 1, pp. 32–43, 2014.
[2] M. V. Kothare, V. Balakrishnan, and M. Morari, “Robust constrained model predictive control using linear matrix inequalities,” Automatica, vol. 32, no. 10, pp. 1361–1379, 1996.
[3] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, “Constrained model predictive control: Stability and optimality,” Automatica, vol. 36, no. 6, pp. 789–814, 2000.
[4] Y. Xi, D. Li, and S. Lin, “Model predictive control - status and challenges,” Acta Automatica Sinica, vol. 39, no. 3, pp. 222–236, 2013.
[5] S. J. Qin, “Survey on data-driven industrial process monitoring and diagnosis,” Annual Reviews in Control, vol. 36, no. 2, pp. 220–234, 2012.
[6] Y. Xu, D. Li, Y. Xi, J. Lan, and T. Jiang, “Improved predictive controller on FPGA by hardware matrix inversion,” IEEE Transactions on Industrial Electronics, 2018.
[7] J. Yang, W. X. Zheng, S. Li, B. Wu, and M. Cheng, “Design of a prediction-accuracy-enhanced continuous-time MPC for disturbed systems via a disturbance observer,” IEEE Transactions on Industrial Electronics, vol. 62, no. 9, pp. 5807–5816, 2015.
[8] H. Shah and M. Gopal, “Model-free predictive control of nonlinear processes based on reinforcement learning,” in IFAC-PapersOnLine, vol. 49, no. 1, 2016, pp. 89–94.
[9] X. Wang, B. Huang, and T. Chen, “Data-driven predictive control for solid oxide fuel cells,” Journal of Process Control, vol. 17, no. 2, pp. 103–114, 2007.
[10] Z. Hou and S. Jin, “Data-driven model-free adaptive control for a class of MIMO nonlinear discrete-time systems,” IEEE Transactions on Neural Networks, vol. 22, no. 12, pp. 2173–2188, 2011.
[11] D. Laury, J. A. Rossiter, J. Sanchis, and M. Martínez, “Data-driven latent-variable model-based predictive control for continuous processes,” Journal of Process Control, vol. 20, no. 10, pp. 1207–1219, 2010.
[12] D. Zhou, D. Sun, Z. Pang, and G. Liu, “Data-based predictive control for networked non-linear systems with two-channel packet dropouts,” IET Control Theory & Applications, vol. 9, no. 7, pp. 1154–1161, 2015.
[13] W. Aangenent, D. Kostic, B. de Jager, R. van de Molengraft, and M. Steinbuch, “Data-based optimal control,” in Proceedings of the Americas Control Conference, 2005, pp. 1460–1465.
[14] S. S. Ge, Z. Li, and H. Yang, “Data-driven adaptive predictive control for holonomic constrained under-actuated hibod robots,” IEEE Transactions on Control Systems Technology, vol. 20, no. 3, pp. 787–795, 2012.
[15] S. S. Ge, C. Yang, S. Member, and T. H. Lee, “Adaptive predictive control using neural network for a class of pure-feedback systems in discrete time,” IEEE Transactions on Neural Networks, vol. 19, no. 9, pp. 1599–1614, 2008.

Fig. 1. Convergence of $A_k$ and $B_k$ to their actual values during the control process.

Fig. 2. The trajectories of the output variables and the flow rates.

$$A = \begin{bmatrix} -17.98 & -295.866 & 0 & 0 & 0 & 0 \\ 0.0207 & 0.1889 & 0.0704 & 0 & 0 & 0 \\ 0 & 0.3879 & 0.8000 & 0 & 0 & 0 \\ 0.0977 & 0 & 0 & -18.01 & -295.87 & 0 \\ 0 & 0.0617 & 0 & 0.0131 & 0.0433 & 0.0589 \\ 0 & 0 & 0 & 0 & 0.3787 & -0.622 \end{bmatrix} , \quad B = \begin{bmatrix} 17.8996 & -13.781 \\ -0.0131 & 0.0101 \\ 17.8636 & 17.8636 \\ 0.0082 & 0.0082 \\ 0 & 0 \end{bmatrix}$$
[16] R. Kadali, B. Huang, and A. Rossiter, “A data driven subspace approach to predictive controller design,” *Control Engineering Practice*, vol. 11, no. 3, pp. 261–278, 2003.

[17] Y. Zhou, D. Li, Y. Xi, and Z. Gan, “Periodic event-triggered control for distributed networked multiagents with asynchronous communication: A predictive control approach,” *International Journal of Robust and Nonlinear Control*, vol. 29, no. 1, pp. 43–66, 2019.

[18] G. Shi and R. E. Skelton, “Markov data-based lqg control,” *Journal of dynamic systems, measurement, and control*, vol. 122, no. 3, pp. 551–559, 2000.

[19] D. Li, Y. Xi, and F. Gao, “Synthesis of dynamic output feedback RMPC with saturated inputs,” *Automatica*, vol. 49, no. 4, pp. 949–954, 2013.

[20] Y. Zhou, D. Li, Y. Xi, and Z. Gan, “Synthesis of model predictive control based on data-driven learning,” *Science China Information Sciences*, vol. 63, pp. 1–3, 2020.

[21] D. Görges, “Relations between model predictive control and reinforcement learning,” in *IFAC-PapersOnLine*, vol. 50, no. 1, 2017, pp. 4920–4928.

[22] Y. Jiang and Z. P. Jiang, “Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics,” *Automatica*, vol. 48, no. 10, pp. 2699–2704, 2012.

[23] ———, *Robust Adaptive Dynamic Programming*. USA: Wiley-IEEE Press, 2017.

[24] Y. Zhou, K. G. Vamvoudakis, W. M. Haddad, and Z.-P. Jiang, “A secure control learning framework for cyber-physical systems under sensor and actuator attacks,” *IEEE Transactions on Cybernetics*, 2020.

[25] ———, “A secure control learning framework for cyber-physical systems under sensor attacks,” in *2019 American Control Conference (ACC)*. IEEE, 2019, pp. 4280–4285.

[26] Y. Zhou, D. Li, J. Lu, Y. Xi, and L. Cen, “Networked and distributed predictive control of non-linear systems subject to asynchronous communication,” *IET Control Theory & Applications*, vol. 12, no. 4, pp. 504–514, 2018.

[27] F. L. Lewis, D. Vrabie, and V. L. Syrmos, *Optimal control*. John Wiley & Sons, 2012.

[28] P. Ioannou and J. Sun, *Robust adaptive control*. Upper Saddle River, NJ: PTR Prentice-Hall, 1996.

[29] Y. Cao and Z. Yang, “Multiobjective process controllability analysis,” *Computer and Chemical Engineering*, vol. 28, pp. 83–90, 2004.

[30] T. Wang, H. Gao, and J. Qiu, “A combined fault-tolerant and predictive control for network-based industrial processes,” *IEEE Transactions on Industrial Electronics*, vol. 63, no. 4, pp. 2529–2536, 2016.