mathematics

MDPI

Article

Economic Machine-Learning-Based Predictive Control of Nonlinear Systems

Zhe Wu [1] **and Panagiotis D. Christofides** [1,2,*]

[1] Department of Chemical and Biomolecular Engineering, University of California, Los Angeles, CA 90095-1592, USA; wuzhe@ucla.edu

[2] Department of Electrical and Computer Engineering, University of California, Los Angeles, CA 90095-1592, USA

[*] Correspondence: pdc@seas.ucla.edu

Received: 2 May 2019; Accepted: 28 May 2019; Published: 1 June 2019

Abstract: In this work, a Lyapunov-based economic model predictive control (LEMPC) method is developed to address economic optimality and closed-loop stability of nonlinear systems using machine learning-based models to make predictions. Specifically, an ensemble of recurrent neural network (RNN) models via a *k*-fold cross validation is first developed to capture process dynamics in an operating region. Then, the LEMPC using an RNN ensemble is designed to maintain the closed-loop state in a stability region and optimize process economic benefits simultaneously. Parallel computing is employed to improve computational efficiency of real-time implementation of LEMPC with an RNN ensemble. The proposed machine-learning-based LEMPC method is demonstrated using a nonlinear chemical process example.

Keywords: economic model predictive control; recurrent neural networks; ensemble learning; nonlinear systems; parallel computing

1. Introduction

Economic model predictive control (EMPC) that integrates optimization of process economics with process control has attracted significant attention over the last decade (e.g., [1–3]). Unlike the traditional two-layered real-time optimization (RTO) that continuously optimizes process set-points in the upper level and applies a tracking model predictive controller or a proportional-integral-derivative controller in the lower layer, EMPC operates the system in a time-varying fashion (off steady-state) by dynamically optimizing an economic cost function accounting for stability constraints in one layer. Many research works have been done on EMPC in terms of closed-loop stability, economic optimality, model uncertainty and process operational safety (e.g., [4–6]). However, to achieve a desired closed-loop performance under EMPC, it is acknowledged that a reliable process model that can predict the future state evolution accurately is a key requirement. To that end, multiple linear state-space models have been used within EMPC to capture the nonlinear dynamics in the operating region in [7]. Additionally, a well-conditioned polynomial nonlinear state-space model has been used in [8] to approximate the actual (first-principles) nonlinear process model (normally substituting of the actual nonlinear process) for EMPC. Given the recent interest in machine learning modeling and particularly its usage in solving nonlinear regression problems ([9,10]), in this work, we will derive a nonlinear data-driven process model for EMPC by taking advantage of machine learning techniques.

Machine learning has been widely used in a variety of engineering problems such as classification, pattern recognition, and numerical prediction. Among machine learning methods, neural networks have shown great success in regression problems. With the rapid development of neural network libraries and application programming interfaces e.g., Tensorflow [11], Caffe [12], Keras [13], etc.), building a neural network has presently become intuitive and straightforward. Additionally, neural network-based modeling outperforms other data-driven modeling methods due to its ability to implicitly detect complex nonlinearities and the availability of multiple training algorithms ([14]). Neural network has undergone several phases in which different theoretical approaches led to increasing understanding of its power and limitations. For example, in the 1980s, the first RNN (i.e., Hopfield networks) was created for pattern recognition ([15]). At that time, implementing a neural network with many layers for complex systems was a technical challenge [16]. Subsequently, deep learning was proposed to alleviate the problems of vanishing gradient in RNNs, which provided a new perspective to the machine learning field [17]. Compared to feedfoward neural networks, RNN is able to memorize previous information in the network and thus, can be applied to model a nonlinear dynamical system from time-series data. However, since the majority of research is focused on validating the identified system, it is not clear how RNN models can be incorporated in EMPC with guaranteed closed-loop stability.

In addition, ensemble learning, a machine learning paradigm that trains multiple models for the same problem, is gaining increasing attention in recent years ([18,19]). By taking advantage of multiple learning algorithms or various training datasets, ensemble learning provides better predictive performance than a single machine learning model. However, the consequent challenge for implementations of ensemble regression models in process control is how to solve the ensemble of RNNs efficiently and in a timely fashion. Although reducing the computation time of neural network training process is always an important research topic in machine learning community, process engineers are more concerned about the real-time implementation of neural networks in process control systems. To address computational implementation issues, parallel computing ([20]) is employed to solve multiple RNN models simultaneously in EMPC.

Motivated by the above, a Lyapunov-based EMPC method using an ensemble of RNN models is developed in this work. Specifically, in Section 2, the notation, the class of nonlinear systems considered and the stabilizability assumptions are introduced. In Section 3, the class of RNN models and the learning algorithm are provided. Then, ensemble learning is used to obtain multiple RNN models based on a k-fold cross validation, while parallel computing is used to improve computational efficiency. Subsequently, in Section 4, the Lyapunov-based EMPC that incorporates an RNN ensemble is formulated to guarantee the boundedness of the state in the closed-loop stability region accounting for sufficiently small modeling error, sampling time and bounded disturbances. In the last section, the proposed LEMPC method is applied to a chemical process example to demonstrate its guaranteed closed-loop stability and economic optimality properties.

2. Preliminaries

2.1. Notation

Throughout the manuscript, the operator $|\cdot|$ denotes the Euclidean norm of a vector. x^T is used to denote the transpose of x. The function $f(\cdot)$ is of class $\mathcal{C}^1$ if it is continuously differentiable in its domain. The notation $L_f V(x)$ denotes the standard Lie derivative $L_f V(x) := \frac{\partial V(x)}{\partial x} f(x)$. "$\backslash$" is used to denote the set subtraction, i.e., $A \backslash B := \{x \in \mathbf{R}^n \mid x \in A, x \notin B\}$. A continuous function $\alpha : [0, a) \to [0, \infty)$ is said to belong to class $\mathcal{K}$ if $\alpha(0) = 0$ and it is strictly increasing.

2.2. Class of Systems

We consider the class of continuous-time nonlinear systems that is described by the following system of first-order nonlinear ordinary differential equations:

$$\dot{x} = F(x, u, w) := f(x) + g(x)u + h(x)w, \; x(t_0) = x_0 \tag{1}$$

where the state vector is $x \in \mathbf{R}^n$, the manipulated input vector is $u \in \mathbf{R}^m$, and the disturbance vector is $w \in W$ with $W := \{w \in \mathbf{R}^q \mid |w| \leq w_m,\ w_m \geq 0\}$. The control action constraint is defined by $u \in U := \{u_i^{\min} \leq u_i \leq u_i^{\max}, i = 1, ..., m\} \subset \mathbf{R}^m$. It is assumed that $f(\cdot) : \mathbf{R}^n \to \mathbf{R}^n$, $g(\cdot) : \mathbf{R}^n \to \mathbf{R}^{n\times m}$, and $h(\cdot) : \mathbf{R}^n \to \mathbf{R}^{n\times q}$ are sufficiently smooth vector and matrix functions. Additionally, it is assumed that the initial time t_0 is zero (i.e., $t_0 = 0$) and $f(0) = 0$. Therefore, a steady-state of the nominal system of Equation (1) (i.e., $w(t) \equiv 0$) is at the origin.

A stabilizing control law $u = \Phi(x) \in U$ that renders the origin of the nominal system of Equation (1) (i.e., $w(t) \equiv 0$) exponentially stable is assumed to exist, in the sense that there exists a $\mathcal{C}^1$ Control Lyapunov function $V(x)$ such that the following inequalities hold for all x in an open neighborhood D around the origin:

$$c_1|x|^2 \leq V(x) \leq c_2|x|^2, \tag{2a}$$

$$\frac{\partial V(x)}{\partial x} F(x, \Phi(x), 0) \leq -c_3|x|^2, \tag{2b}$$

$$\left|\frac{\partial V(x)}{\partial x}\right| \leq c_4|x| \tag{2c}$$

where c_1, c_2, c_3 and c_4 are positive constants. $F(x, u, w)$ is the nonlinear system of Equation (1). Based on the universal Sontag control law ([21]), a candidate controller $\Phi(x)$ is given by the saturated control law accounting for the input constraint $u \in U$.

The stability region for the closed-loop system of Equation (1) is characterized as a level set of Lyapunov function $V(x)$ within D where the time-derivative of the Lyapunov function V is negative. Specifically, based on Equation (2), we can first find a region ϕ_u where $\dot{V}(x)$ is rendered negative under the controller $u = \Phi(x) \in U$: $\phi_u := \{x \in \mathbf{R}^n \mid \dot{V}(x) = L_f V + L_g V u < -kV(x), u = \Phi(x) \in U\} \cup \{0\}$, with $k > 0$. Then the closed-loop stability region Ω_ρ is characterized as a subset in ϕ_u: $\Omega_\rho := \{x \in \phi_u \mid V(x) \leq \rho\}$, where $\rho > 0$. Since $\Omega_\rho \subset \phi_u$, it follows that Ω_ρ is a forward invariant set. Additionally, the Lipschitz property of $F(x, u, w)$ combined with the bounds on u and w implies that there exist positive constants M, L_x, L_w, $L_x^{'}$, $L_w^{'}$ such that the following inequalities hold for all $x, x' \in D, u \in U$, and $w \in W$:

$$|F(x, u, w)| \leq M \tag{3a}$$

$$|F(x, u, w) - F(x', u, 0)| \leq L_x|x - x'| + L_w|w| \tag{3b}$$

$$\begin{aligned}&\left|\frac{\partial V(x)}{\partial x} F(x, u, w) - \frac{\partial V(x')}{\partial x} F(x', u, 0)\right| \\ &\leq L_x^{'}|x - x'| + L_w^{'}|w|\end{aligned} \tag{3c}$$

Remark 1. *Since closed-loop stability is achieved for the nonlinear system of Equation (1) under the controller $u = \Phi(x) \in U$ for all initial conditions $x_0 \in \Omega_\rho$ ([4]), the dataset for training recurrent neural networks will be generated within the closed-loop stability region Ω_ρ such that there exist feasible control actions $u \in U$ for the resulting recurrent neural network that captures the nonlinear dynamics in Ω_ρ.*

3. Recurrent Neural Network

In this work, we develop a nonlinear recurrent neural network (RNN) model with the following form:

$$\dot{\hat{x}} = F_{nn}(\hat{x}, u) := A\hat{x} + \Theta^T y \tag{4}$$

where $\hat{x} \in \mathbf{R}^n$ is the RNN state vector and $u \in \mathbf{R}^m$ is the manipulated input vector. $y = [y_1, ..., y_n, y_{n+1}, ..., y_{m+n}] = [\sigma(\hat{x}_1), ..., \sigma(\hat{x}_n), u_1, ..., u_m] \in \mathbf{R}^{n+m}$ is a vector of both the network state $\hat{x}$ and the input u, where $\sigma(\cdot)$ is the nonlinear activation function (e.g., a sigmoid function $\sigma(x) = 1/(1 + e^{-x})$). A is a diagonal matrix, i.e., $A = diag\{-a_1, ..., -a_n\} \in \mathbf{R}^{n\times n}$, and Θ is a coefficient matrix $\Theta = [\theta_1, ..., \theta_n] \in \mathbf{R}^{(m+n)\times n}$ with $\theta_i = b_i[w_{i1}, ..., w_{i(m+n)}]$, $i = 1, ..., n$. a_i and b_i are constants, where a_i is assumed to be positive such that each state $\hat{x}_i$ is bounded-input bounded-state stable. w_{ij} is the weight connecting the jth input to the ith neuron where $i = 1, ..., n$ and $j = 1, ..., (m+n)$. Since the RNN model of Equation (4) is affine in the manipulated inputs u, the RNN model can also be represented in a similar form of Equation (1) by $\dot{\hat{x}} = \hat{f}(\hat{x}) + \hat{g}(\hat{x})u$, where $\hat{f}(\cdot)$ and $\hat{g}(\cdot)$ are derived from coefficients A and Θ in Equation (4). Similarly, $\hat{f}(\cdot)$ and $\hat{g}(\cdot)$ are assumed to be sufficiently smooth vector and matrix functions of dimensions $n \times 1$ and $n \times m$, respectively.

It is shown in Figure 1 that RNNs possess a certain type of memory by introducing feedback loops in the hidden layer such that the states information derived from earlier inputs remains in the network. Additionally, if we unfold one of the RNN states in time as shown in Figure 1, the manipulated input vector u_t and the previous state vector x_{t-1} are fed into the current state x_t with the weight matrix Θ, where x_{t-1}, x_t, and x_{t+1} are all internal states of an RNN model.

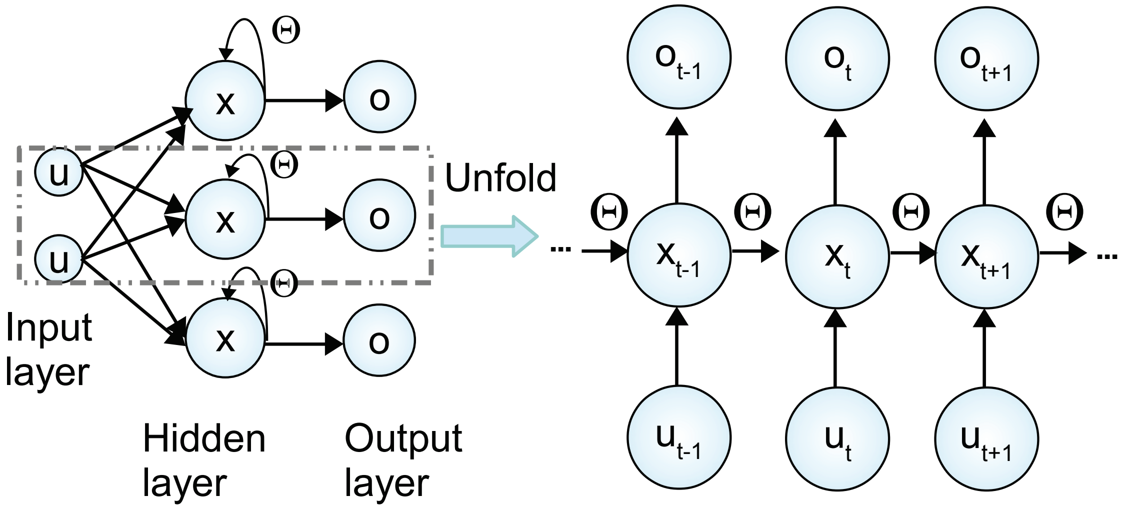

Figure 1. A recurrent neural network and its unfolded structure, where Θ is the weight matrix, x is the state vector, u is the manipulated input vector and o is the output vector.

Due to the capability of capturing nonlinear dynamic behavior and the universal approximation theorem ([22,23]) demonstrating that the RNN model with sufficient neurons is able to approximate any dynamic nonlinear system on compact subsets of the state-space for finite time, the RNN model of Equation (4) is used to approximate the continuous-time nonlinear system of Equation (1) in this work.

3.1. RNN Learning Algorithm

The RNN learning algorithm is developed to obtain the optimal weight matrix Θ^*, under which the modeling error between the nominal system of Equation (1) (i.e., $w(t) \equiv 0$) and the the RNN model of Equation (4) is minimized. The modeling error $\nu := F(x, u, 0) - F_{nn}(\hat{x}, u)$ under $\Theta = \Theta^*$ is assumed to be nonzero since the obtained RNN model may not perfectly approximate the nominal system of Equation (1) due to insufficient number of nodes or layers. Additionally, the modeling error $\nu(t)$ is bounded (i.e., $|\nu(t)| \leq \nu_m$, $\nu_m > 0$) since the RNN model of Equation (4) is developed based on the dataset generated within Ω_ρ where $x(t)$ and $u(t) \in U$ are bounded. The weight vector θ_i of the RNN model of Equation (4) is also bounded by $|\theta_i| \leq \theta_m$, with $\theta_m > 0$ to prevent the weights from drifting to infinity. Throughout the manuscript, we will use $x(t)$ to represent the state of the nonlinear system of Equation (1) and $\hat{x}(t)$ to represent the state of the RNN model of Equation (4).

According to the definition of the modeling error ν, the nominal system of Equation (1) (i.e., $w(t) \equiv 0$) can be written in the following form:

$$\dot{x}_i = -a_i x_i + \theta_i^{*T} y + \nu_i,\ i = 1, ..., n \tag{5}$$

where the optimal weight vector θ_i^* is obtained as follows:

$$\theta_i^* := \arg \min_{|\theta_i| \le \theta_m} \{ \sum_{k=1}^{N_d} |F_i(x_k, u_k, 0) + a_i x_k - \theta_i^T y_k| \} \tag{6}$$

where N_d is the number of data samples used for training. Consider the state error $e = \hat{x} - x \in \mathbf{R}^n$. Based on Equations (4) and (5), the time-derivative of the state error is derived as follows:

$$\dot{e}_i = \dot{\hat{x}}_i - \dot{x}_i = -a_i e_i + \zeta_i^T y - \nu_i,\ i = 1, ..., n \tag{7}$$

where $\zeta_i = \theta_i - \theta_i^*$ is the error between the current weight vector θ_i and the unknown optimal weight vector θ_i^*. The weight vector θ is updated by the following equation:

$$\dot{\theta}_i = -\eta_i y e_i,\ i = 1, ..., n \tag{8}$$

where the learning rate η_i is a positive real number. Based on the learning law of Equation (8), the following theorem is established to demonstrate that the state error e remains bounded and is constrained by the modeling error $|\nu|$.

Theorem 1. *Consider the RNN model of Equation (4) with the weights updating law of Equation (8). In the presence of a nonzero modeling error ν, the state error e_i and the weight error ζ_i are bounded, and there exist $\lambda \in \mathbf{R}$ and $\mu > 0$ such that the following inequality holds:*

$$\int_0^t |e(\tau)|^2 d\tau \le \lambda + \mu \int_0^t |\nu(\tau)|^2 d\tau \tag{9}$$

Proof. We first define a Lyapunov function $\tilde{V} = \frac{1}{2} \sum_{i=1}^n (e_i^2 + \zeta_i^T \eta_i^{-1} \zeta_i)$ and prove the boundedness of e_i and ζ_i by showing that $\tilde{V}$ is bounded. Specifically, using Equations (7) and (8) and the fact that $\dot{\zeta}_i = \dot{\theta}_i$, the time-derivative of $\tilde{V}$ is first derived as follows:

$$\begin{aligned} \dot{\tilde{V}} &= \sum_{i=1}^n (e_i \dot{e}_i + \eta_i^{-1} \zeta_i \dot{\zeta}_i) \\ &= \sum_{i=1}^n (-a_i e_i^2 - e_i \nu_i) \end{aligned} \tag{10}$$

In the presence of modeling error $\nu_i \neq 0$, it is observed that $\dot{\tilde{V}} \le 0$ does not hold for all times. Therefore, let $\alpha := \min\{a_i, 1/(\lambda_m),\ i = 1, ..., n\}$ and $\beta := \sum_{i=1}^n (2\theta_m^2 + \nu_m^2 / 2a_i)$, where λ_m represents the maximum eigenvalue of η_i^{-1}. The following inequality is further derived based on Equation (10):

$$\begin{aligned} \dot{\tilde{V}} &= \sum_{i=1}^n (-\frac{a_i}{2} e_i^2 - \frac{1}{2} |\zeta_i|^2) + (\frac{1}{2} |\zeta_i|^2 - \frac{a_i}{2} e_i^2 - e_i \nu_i) \\ &\le -\alpha \tilde{V} + \sum_{i=1}^n (\frac{1}{2} |\zeta_i|^2 + \frac{1}{2a_i} \nu_i^2) \end{aligned} \tag{11}$$

Since the weight vector is bounded by $|\theta_i| \le \theta_m$, it is derived that $\frac{1}{2} |\zeta_i|^2 \le 2\theta_m^2$. According to the definitions of α and β, it follows that $\dot{\tilde{V}} \le -\alpha \tilde{V} + \beta$. Therefore, for all $\tilde{V} \ge V_0 =$

β/α, $\dot{\tilde{V}} \leq 0$ holds. This implies that $\tilde{V}$ is bounded, and thus, e_i and ζ_i are bounded as well. Additionally, from Equation (10), it is readily shown that the following equations hold:

$$\begin{aligned}
\dot{\tilde{V}} &= \sum_{i=1}^{n}(-\frac{a_i}{2}e_i^2 - \frac{a_i}{2}e_i^2 - e_i\nu_i) \\
&= \sum_{i=1}^{n}(-\frac{a_i}{2}e_i^2 - (\frac{a_i}{2}e_i^2 + e_i\nu_i + \frac{1}{2a_i}\nu_i^2) + \frac{1}{2a_i}\nu_i^2) \\
&\leq \sum_{i=1}^{n}(-\frac{a_i}{2}e_i^2 + \frac{1}{2a_i}\nu_i^2)
\end{aligned} \tag{12}$$

From Equation (12), $\tilde{V}(t)$ is obtained as follows:

$$\begin{aligned}
\tilde{V}(t) &\leq \tilde{V}(0) + \sum_{i=1}^{n}(-\frac{a_i}{2}\int_0^t e_i(\tau)^2 d\tau + \frac{1}{2a_i}\int_0^t \nu_i(\tau)^2 d\tau) \\
&\leq \tilde{V}(0) - \frac{a_{min}}{2}\int_0^t |e(\tau)|^2 d\tau + \frac{1}{2a_{min}}\int_0^t |\nu(\tau)|^2 d\tau
\end{aligned} \tag{13}$$

where a_{min} is the minimum value of a_i, $i = 1, ..., n$. Let $\lambda = \frac{2}{a_{min}} \sup_{t\geq 0}(\tilde{V}(0) - \tilde{V}(t))$ and $\mu = 1/a_{min}^2$. The following inequality is derived:

$$\begin{aligned}
\int_0^t |e(\tau)|^2 d\tau &\leq \frac{2}{a_{min}}(\tilde{V}(0) - \tilde{V}(t)) + \frac{1}{a_{min}^2}\int_0^t |\nu(\tau)|^2 d\tau \\
&\leq \lambda + \mu \int_0^t |\nu(\tau)|^2 d\tau
\end{aligned} \tag{14}$$

The above inequality shows that the state error e is bounded and is proportional to the modeling error $|\nu|$ between the RNN model of Equation (4) and the nominal system of Equation (1) (i.e., $w(t) \equiv 0$). Furthermore, if there exists a constant $C > 0$ such that $\int_0^\infty |\nu(t)|^2 dt$ is bounded (i.e., $\int_0^\infty |\nu(t)|^2 dt = C < \infty$), then it follows that $\int_0^\infty |e(t)|^2 dt \leq \lambda + \mu C < \infty$. Therefore, $e(t)$ converges to zero asymptotically since $e(t)$ is a uniformly continuous function implied by the boundedness of its derivative $\dot{e}$. □

3.2. *Ensemble of RNN Models*

In this section, an RNN model is initially developed to approximate the nonlinear system of Equation (1) within an operating region. Based on that, the ensemble learning using the aforementioned RNN learning algorithm is used with a k-fold cross validation to improve prediction accuracy of RNN models.

3.2.1. Development of RNN Models

The development of RNN models consists of two steps: data generation and training process. In general, the dataset for training an RNN model is a collection of sampled data from industrial processes, laboratory experiments, or simulations of a given system. There are many high-quality training datasets, for example, the collection of databases from the UCI Machine Learning Repository [24] that is widely used by the researchers in machine learning community. In this work, we use the simulation data of the nominal system of Equation (1) (i.e., $w(t) \equiv 0$) as the training and validation datasets for the RNN model of Equation (4). Specifically, to generate the dataset that captures the system dynamics for $x \in \Omega_\rho$ and $u \in U$, open-loop simulations are first conducted for initial conditions $x_0 \in \Omega_\rho$ with the constrained input vector $u \in U$. It should be mentioned that the continuous system of Equation (1) is simulated in a sample-and-hold fashion (i.e., the input is a piecewise constant function, i.e., $u(t) = u(t_k)$, $\forall t \in [t_k, t_{k+1})$, where $t_{k+1} := t_k + \Delta$ and Δ is the sampling period). Then, the nominal system of Equation (1) is integrated via explicit Euler method with a sufficiently small integration time step $h_c < \Delta$. The dataset is generated with the following steps: (1) The targeted region Ω_ρ and the range of inputs $u \in U$ are first discretized with sufficiently small intervals. (2) Open-loop simulations of the nominal system of Equation (1) are performed in a sample-and-hold fashion with various initial states $x_0 \in \Omega_\rho$ and inputs $u \in U$ to obtain a large number of state trajectories for a fixed length of time to sweep over all the values that (x, u) can take. It is noted that data points are collected at every integration time step h_c instead of every sampling step Δ to obtain intermediate states as many as possible in order to approximate a continuous system of Equation (1). (3) Then, the obtained state trajectories are separated into many time-series data samples with the length of P_{nn}, which represents the prediction period for RNN models. In other words, the RNN model is developed to predict future states over $t \in [t_k, t_k + P_{nn}]$, where all the data points between t_k and $t_k + P_{nn}$ are treated as the internal states of the RNN model with the time interval of h_c (i.e., the time interval between two consecutive states x_{t-1} and x_t for the unfolded RNN shown in Figure 1 is equal to h_c). (4) The dataset is partitioned into training, validation and testing datasets.

Next, the RNN model is developed based on the dataset generated from open-loop simulations via an application program interface (API), i.e., Keras ([13]). To guarantee that the RNN model of Equation (4) achieves good performance in a neighborhood around the origin and has the same equilibrium point as the nonlinear system of Equation (1), the modeling error is further required to be bounded by $|\nu| \leq \gamma|x| \leq \nu_m$ during the training process. Before starting an RNN training process, the number of layers and nodes used, the optimizer, and the type of loss function need to be specified at the beginning. Specifically, in our work, the number of layers and nodes for RNNs are determined through a grid search and the adaptive moment estimation method (i.e., Adam in Keras) is used as RNN optimizer. The loss function is the mean absolute percentage error between predicted states $\hat{x}$ and actual states x from training data. Additionally, to prevent RNNs from over-training, the early stopping criteria is employed to terminate the training process before the maximum number of epochs is reached if the following two conditions are satisfied: (1) The modeling error is below its threshold. (2) The performance on the validation dataset stops improving.

Remark 2. *It is noted that original data samples from open-loop simulations are shortened to a fixed length, P_{nn}, in the step 3 of data generation above. Through this data preprocessing step, the dataset is applicable for developing an RNN with the prediction period P_{nn}, which is chosen to be an integer multiple of the sampling period Δ such that the RNN model of Equation (4) is able to predict future states for at least one sampling period. One of the advantages of introducing step 2 and step 3 is that a new dataset can be conveniently generated for an RNN with a different prediction period P_{nn} while the original dataset in step 2 only needs to be created once.*

Remark 3. *Unlike feedforward neural networks (FNN) that take state measurement $x(t_k)$ and manipulated inputs $u(t_k)$ at time $t = t_k$ to predict the state at $t = t_k + P_{nn}$ only, RNNs use all the intermediate states between $x(t_k)$ and $x(t_k + P_{nn})$ to predict the state trajectory over this prediction period. In other words, all the internal states with the time step h_c are used as one of the inputs (the other one is the manipulated input u) to the RNN (shown in Figure 1) over $t \in [t_k, t_k + P_{nn}]$ during the training process. Therefore, besides the final predicted state $x(t_k + P_{nn})$, all the internal states before can be predicted and exported (if needed) by RNNs. Additionally, it is important to mention that since the time interval for RNN internal states is chosen to be the sufficiently small integration time step h_c that is used to simulate the continuous-time nonlinear system of Equation (1), the obtained RNN model can be regarded as a good representation for the continuous-time RNN model of Equation (4).*

Remark 4. *Data quality plays an important role in data-driven modeling techniques including RNN modeling. In case a good RNN model is initially not available due to poor data quality, the RNN model can be improved via on-line update through real-time process state measurements. Additionally, if the real-time process measurements are of poor quality, for example, noisy measurements, a data preprocessing step can be employed before training an RNN. Overall, creating a good dataset remains an important issue in RNN modeling; in the chemical reactor example in Section 5 the data is generated from open-loop simulations.*

Remark 5. *The RNN models considered in our work involve the use of nonlinear terms in the dynamic model (this can be thought of as a nonlinear process gain $k(x)$, multiplying the input), which allows capturing gain sign change in the different regions in the operating state-space if it occurs. This issue does not occur in the operating region of the chemical reactor example considered in Section 5. The reader may refer to [25] for an approach to neural network modeling that captures gain change.*

3.2.2. Ensemble Learning

Ensemble learning is a machine learning process that combines multiple learning algorithms to obtain better prediction performance ([26]) in terms of reduced variability and improved generalization performance. The reasons that ensemble regression models are able to improve prediction performance are given in [10,26]. In this work, homogeneous ensemble regression models are constructed following the RNN learning algorithm in Section 3.1 and a k-fold cross validation. Specifically, as shown in Figure 2, the dataset is initially separated into k folds with the same size, where $k-1$ folds are used for training and the remaining one is used for validation. By choosing a different subset as the validation dataset every time, a total number of k different RNNs can be obtained. Subsequently, the final prediction results of an ensemble of RNN models is calculated using the stacking method that averages the predictions of multiple RNN models. It is noted that besides the stacking method, other ensemble-based algorithms such as Boosting and Bagging [19] can also be applied to improve the overall prediction performance within the targeted operating region.

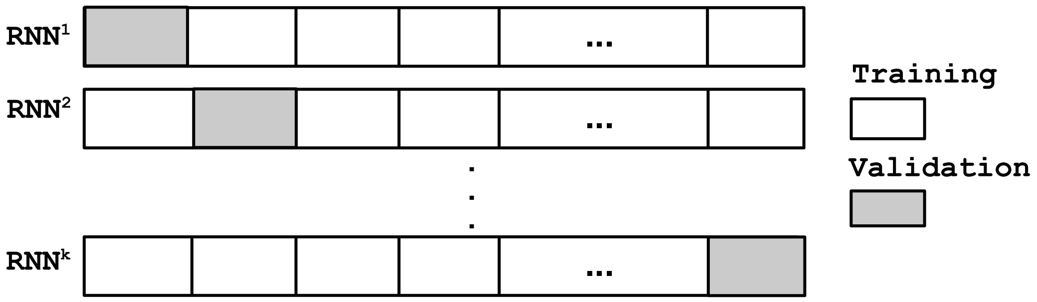

Figure 2. An ensemble of k RNN models based on a k-fold cross validation.

3.2.3. Parallel Computing

Parallel computing is a type of computation in which the execution of multiple processes or of the computational tasks are carried out simultaneously. Since the final prediction results in ensemble learning are obtained by combining the results of individual RNN models that are independent from each other, the computational tasks for all RNN models can be operated in parallel to improve computational efficiency. The algorithm for parallel computation of an ensemble of k RNN models is shown in Figure 3. The blocks on the left demonstrate the state prediction using parallel computing and the dashed blocks on the right represent the actual nonlinear system of Equation (1) and the model-based controller that will be discussed in the next section. Specifically, we first reserve $k+1$ nodes, where node 0 is the host node and the rest are worker nodes. The host node is used to exchange information while the worker nodes are mainly used for computation. Each worker node is assigned with a corresponding RNN model for prediction. At time $t = t_k$, the model-based controller receives the state measurement $x(t_k)$ and sends it to the host node 0 along with the manipulated inputs $u^g(t_k)$ (i.e., the guess of control actions over the prediction horizon). The host node broadcasts $x(t_k)$ and $u^g(t_k)$ to all worker nodes such that all RNN models share the same input vector. Prediction of RNN models are carried out in all worker nodes simultaneously. It is noted that a synchronization barrier is placed before the ensemble combiner in node 0 to ensure that all the computation tasks in worker nodes are completed. Lastly, the host node combines prediction results from multiple RNN models to obtain predicted states over $t \in [t_k, t_k + Pnn]$ and sends them to the model-based controller. The optimal control actions $u^*(t_k)$ is calculated by the optimizer solver in model-based controller by taking various guesses of control actions $u^g(t_k)$, and will be sent to the nonlinear process of Equation (1) to be applied for the next sampling period. The above process is repeated as the horizon of the nonlinear process of Equation (1) is rolled one sampling period forward each time.

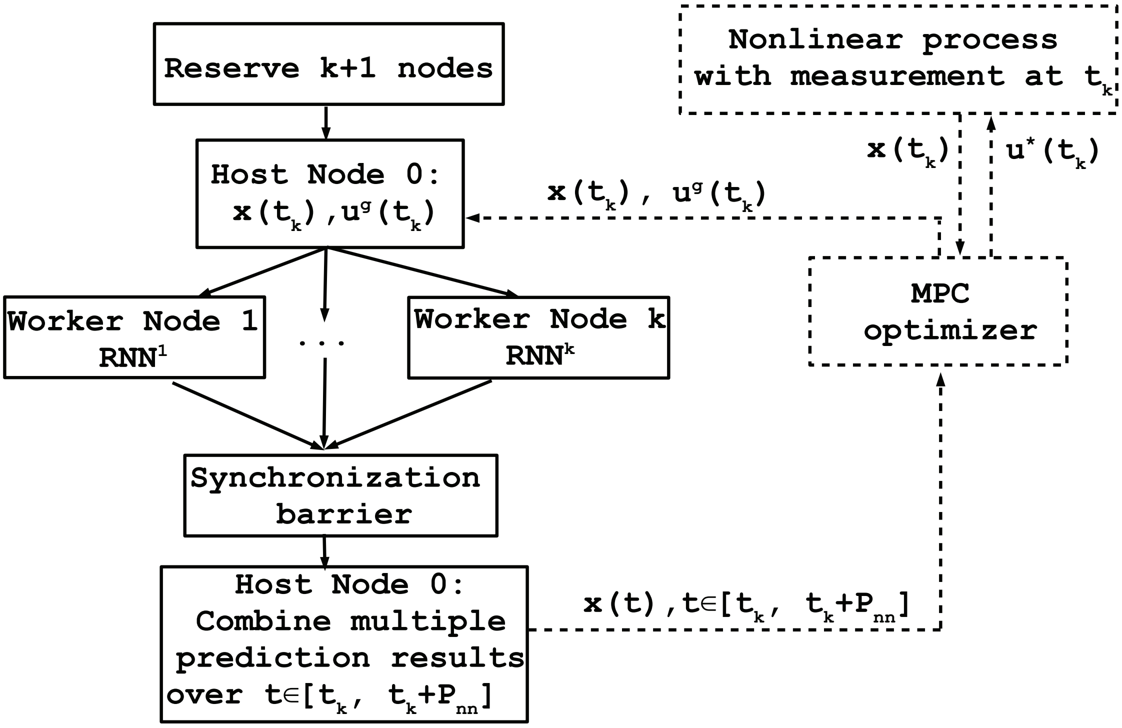

Figure 3. Algorithm for parallel computation of an ensemble of k RNN models, where $x(t_k)$ is the state measurement, $u^g(t_k)$ is the initial guess of manipulated inputs provided by the MPC optimizer, and $u^*(t_k)$ is the optimal control action sent to the nonlinear process of Equation (1), respectively.

4. Lyapunov-Based EMPC Using Ensemble RNN Models

In this section, the Lyapunov-based economic MPC (LEMPC) that incorporates an ensemble of RNN models to predict future states is developed for the nonlinear system of Equation (1). In the following subsections, a Lyapunov-based controller using the RNN model of Equation (4) is first used to characterize the closed-loop stability region in which the origin of the nonlinear system of Equation (1) can be rendered exponentially stable. Then, the formulation of the LEMPC using an ensemble of RNN models is given and Theorem 2 is established to demonstrate closed-loop stability for all initial conditions in the closed-loop stability region.

4.1. Lyapunov-Based Control Using RNN Models

We assume that there exists a stabilizing feedback controller $u = \Phi_{nn}(x) \in U$ that can render the origin of the RNN model of Equation (4) exponentially stable in an open neighborhood $\hat{D}$ in the sense that there exists a C^1 Lyapunov function $\hat{V}(x)$ such that the following inequalities hold for all x in $\hat{D}$:

$$\hat{c}_1|x|^2 \leq \hat{V}(x) \leq \hat{c}_2|x|^2, \tag{15a}$$

$$\frac{\partial \hat{V}(x)}{\partial x} F_{nn}(x, \Phi_{nn}(x)) \leq -\hat{c}_3|x|^2, \tag{15b}$$

$$\left|\frac{\partial \hat{V}(x)}{\partial x}\right| \leq \hat{c}_4|x| \tag{15c}$$

where $\hat{c}_1$, $\hat{c}_2$, $\hat{c}_3$ and $\hat{c}_4$ are positive constants. Although the RNN model of Equation (4) is obtained based on the dataset within the closed-loop stability region Ω_ρ for the nonlinear system of Equation (1), it is noted that the existence a feasible control action $u \in U$ that can render the origin of the RNN model of Equation (4) exponentially stable is not guaranteed due to model mismatch. Therefore, based on Equation (15), the closed-loop stability region is re-characterized for the RNN model of Equation (4) accounting for the input constraints. Specifically, a Lyapunov-based controller based on the RNN model is first developed using the universal Sontag control law ([21]) since the RNN model of Equation (4) can be represented by $\dot{\hat{x}} = \hat{f}(\hat{x}) + \hat{g}(\hat{x})u$. Then, following the characterization method of Ω_ρ for the nonlinear system of Equation (1), an open set from which the origin of the RNN model of Equation (4) can be rendered exponentially stable under the controller $u = \Phi_{nn}(x) \in U$ is characterized as follows: $\hat{\phi}_u := \{x \in \mathbf{R}^n \mid \dot{\hat{V}}(x) < -k\hat{V}(x), u = \Phi_{nn}(x) \in U\} \cup \{0\}$, where $\dot{\hat{V}}(x) = \frac{\partial \hat{V}(x)}{\partial x} F_{nn}(x, \Phi_{nn}(x))$ and $k > 0$. The closed-loop stability region for the RNN model of Equation (4) is defined as the largest level of $\hat{V}$ inside $\hat{\phi}_u$: $\Omega_{\hat{\rho}} := \{x \in \hat{\phi}_u \mid \hat{V}(x) \leq \hat{\rho}\}$, where $\hat{\rho} > 0$. From now on, $\Omega_{\hat{\rho}}$ will be taken as the new operating region for the closed-loop system of Equation (1) in the following sections. Additionally, the Lipschitz property of $F_{nn}(x, u)$ and the boundedness of u imply that there exist positive constants M_{nn} and L_{nn} such that the following inequalities hold for all $x, x' \in \Omega_{\hat{\rho}}, u \in U$:

$$|F_{nn}(x,u)| \leq M_{nn} \tag{16a}$$

$$\left| \frac{\partial \hat{V}(x)}{\partial x} F_{nn}(x,u) - \frac{\partial \hat{V}(x')}{\partial x} F_{nn}(x',u) \right| \\ \leq L_{nn}|x - x'| \tag{16b}$$

The following proposition is developed to demonstrate that the state of the nonlinear system of Equation (1) can be driven towards the origin exponentially for all $x_0 \in \Omega_{\hat{\rho}}$ under the stabilizing controller $u = \Phi_{nn}(x) \in U$ designed for the RNN model of Equation (4) provided that the modeling error is sufficiently.

Proposition 1. *Consider the stabilizing controller $u = \Phi_{nn}(x) \in U$ that renders the origin of the RNN system of Equation (4) exponentially stable for all $x_0 \in \Omega_{\hat{\rho}}$. If there exists a positive real number γ that constrains the modeling error $|\nu| = |F(x,u,0) - F_{nn}(x,u)| \leq \gamma|x| \leq \nu_m$, $\forall x \in \Omega_{\hat{\rho}}$, $u \in U$ and satisfies the following inequality:*

$$\gamma < \hat{c}_3/\hat{c}_4 \tag{17}$$

then the origin of the nominal closed-loop system of Equation (1) under $u = \Phi_{nn}(x) \in U$ is also exponentially stable for all $x \in \Omega_{\hat{\rho}}$.

Proof. We demonstrate the exponential stability of the origin of the nominal system of Equation (1) under $u = \Phi_{nn}(x) \in U$, $\forall x \in \Omega_{\hat{\rho}}$ by showing that there exists a positive real number k' such that $\dot{\hat{V}}(x) \leq -k'|x|^2$ holds for all $\forall x \in \Omega_{\hat{\rho}}$, where $\dot{\hat{V}}(x) = \frac{\partial \hat{V}(x)}{\partial x} F(x, \Phi_{nn}(x), 0)$. Based on Equations (15b) and (15c), the time-derivative of $\hat{V}$ is derived as follows:

$$\begin{aligned} \dot{\hat{V}} =& \frac{\partial \hat{V}(x)}{\partial x} F(x, \Phi_{nn}(x), 0) \\ =& \frac{\partial \hat{V}(x)}{\partial x} (F_{nn}(x, \Phi_{nn}(x)) + F(x, \Phi_{nn}(x), 0) \\ & - F_{nn}(x, \Phi_{nn}(x))) \\ \leq & -\hat{c}_3|x|^2 + \hat{c}_4|x|(F(x, \Phi_{nn}(x), 0) - F_{nn}(x, \Phi_{nn}(x))) \\ \leq & -\hat{c}_3|x|^2 + \hat{c}_4\gamma|x|^2 \end{aligned} \tag{18}$$

If γ satisfies Equation (17), then there exists a positive real number $\tilde{c}_3 = -\hat{c}_3 + \hat{c}_4\gamma > 0$ such that $\dot{\hat{V}} \leq -\tilde{c}_3|x|^2 \leq 0$ holds. Let $k' = \tilde{c}_3$ and this completes the proof that the closed-loop state of the nominal system of Equation (1) is driven to the origin in an exponential rate of convergence under $u = \Phi_{nn}(x) \in U$ for all $x_0 \in \Omega_{\hat{\rho}}$. □

4.2. LEMPC Using an Ensemble of RNN Models

The Lyapunov-based economic MPC (LEMPC) design using an ensemble of RNN models is represented by the following optimization problem:

$$\mathcal{J} = \max_{u \in S(\Delta)} \int_{t_k}^{t_{k+N}} l_e(\tilde{x}(t), u(t))dt \tag{19a}$$

$$\text{s.t.} \quad \dot{\tilde{x}}(t) = \frac{1}{N_e} \sum_{j=1}^{N_e} F_{nn}^j(\tilde{x}(t), u(t)) \tag{19b}$$

$$u(t) \in U, \; \forall \, t \in [t_k, t_{k+N}) \tag{19c}$$

$$\tilde{x}(t_k) = x(t_k) \tag{19d}$$

$$\begin{aligned} &\hat{V}(\tilde{x}(t)) \leq \hat{\rho}_e, \; \forall \, t \in [t_k, t_{k+N}), \\ &\text{if } x(t_k) \in \Omega_{\hat{\rho}_e} \end{aligned} \tag{19e}$$

$$\begin{aligned} &\dot{\hat{V}}(x(t_k), u) \leq \dot{\hat{V}}(x(t_k), \Phi_{nn}(x(t_k))), \\ &\text{if } x(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\hat{\rho}_e} \end{aligned} \tag{19f}$$

where $S(\Delta)$ is the set of piecewise constant functions with period Δ, $\tilde{x}$ is the predicted state trajectory, N_e is the number of RNN models used in predicting future states, and N is the number of sampling periods in the prediction horizon. We use $\dot{\hat{V}}(x, u)$ to represent the time-derivative of $\hat{V}$, i.e., $\dot{\hat{V}}(x, u) = \frac{\partial \hat{V}(x)}{\partial x}(F_{nn}(x, u))$. The optimization problem of Equation (19) is solved by optimizing the time integral of the stage cost function $l_e(\tilde{x}(t), u(t))$ of Equation (19a) over the prediction horizon accounting for the constraints of Equations (19b)–(19f). The constraint of Equation (19b) is the ensemble of RNN models of Equation (4) that is used as the prediction model. Equation (19c) defines the input constraints that are applied over the entire prediction horizon. Equation (19d) defines the initial condition $\tilde{x}(t_k)$ of Equation (19b). The constraint of Equation (19e) maintains the closed-loop state predicted by Equation (19b) in $\Omega_{\hat{\rho}_e}$ over the prediction horizon if the state $x(t_k)$ is inside $\Omega_{\hat{\rho}_e}$. However, if $x(t_k)$ leaves $\Omega_{\hat{\rho}_e}$ but still remains in $\Omega_{\hat{\rho}}$, the contractive constraint of Equation (19f) drives the state towards the origin for the next sampling period such that the state will eventually enter $\Omega_{\hat{\rho}_e}$ within finite sampling periods. By introducing a conservative region $\Omega_{\hat{\rho}_e}$ with $\hat{\rho}_e < \hat{\rho}$ and the stability constraints of Equations (19e) and (19f), it will be proved in Theorem 2 that for any initial condition $x_0 \in \Omega_{\hat{\rho}}$, the closed-loop state of the nonlinear system of Equation (1) is bounded in $\Omega_{\hat{\rho}}$ for all times. A schematic of $\Omega_{\hat{\rho}}$ and $\Omega_{\hat{\rho}_e}$ is given by Figure 4, in which it is shown that the state trajectory starting inside $\Omega_{\hat{\rho}}$ is bounded in $\Omega_{\hat{\rho}}$ under LEMPC.

The optimal input trajectory is calculated over the entire prediction horizon $t \in [t_k, \, t_{k+N})$ and is denoted by $u^*(t)$. However, only the control action $u^*(t_k)$ for the first sampling period in $t \in [t_k, \, t_{k+N})$ is sent to the nonlinear system of Equation (1) to be applied over the next sampling period. Before we demonstrate closed-loop stability for the nonlinear system of Equation (1) under the LEMPC of Equation (19), the next two propositions are first derived to provide useful tools for proving closed-loop stability in Theorem 2. Specifically, the following proposition provides an upper bound for the state error in the presence of sufficiently small modeling error and bounded disturbances.

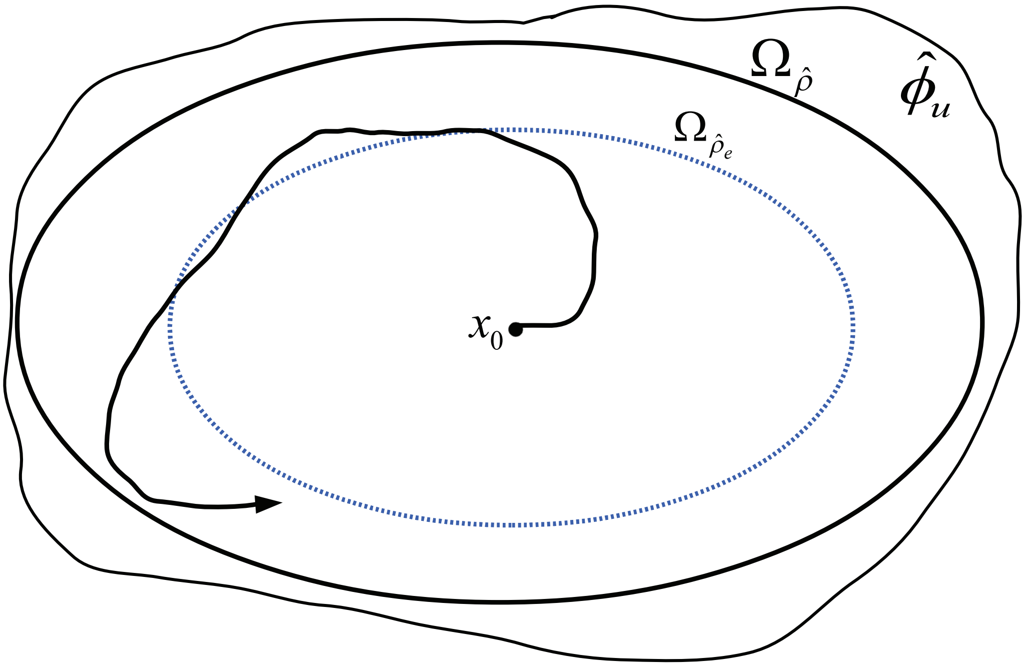

Figure 4. A schematic representing the set $\hat{\phi}_u$, the closed-loop stability region $\Omega_{\hat{\rho}}$, and the set $\Omega_{\hat{\rho}_e}$. Under the LEMPC of Equation (19), the closed-loop state trajectory is bounded in $\Omega_{\hat{\rho}}$ for all times for any $x_0 \in \Omega_{\hat{\rho}}$.

Proposition 2. *Consider the solution $x(t)$ of the nonlinear system $\dot{x} = F(x, u, w)$ of Equation (1) in the presence of bounded disturbances $|w(t)| \leq w_m$, and the solution $\hat{x}(t)$ of the RNN model $\dot{\hat{x}} = F_{nn}(\hat{x}, u)$ of Equation (4) with the same initial condition $x_0 = \hat{x}_0 \in \Omega_{\hat{\rho}}$. If $x(t)$, $\hat{x}(t) \in \Omega_{\hat{\rho}}$, and the modeling error is bounded (i.e., $|\nu(t)| = |\dot{x} - \dot{\hat{x}}| \leq \nu_m$) for all times, then there exists a positive constant κ and a class $\mathcal{K}$ function $f_w(\cdot)$ such that the following inequalities hold $\forall x, \hat{x} \in \Omega_{\hat{\rho}}$ and $w(t) \in W$:*

$$|x(t) - \hat{x}(t)| \leq f_w(t) := \frac{L_w w_m + \nu_m}{L_x}(e^{L_x t} - 1) \tag{20a}$$

$$\hat{V}(x) \leq \hat{V}(\hat{x}) + \frac{\hat{c}_4\sqrt{\hat{\rho}}}{\sqrt{\hat{c}_1}}|x - \hat{x}| + \kappa|x - \hat{x}|^2 \tag{20b}$$

Proof. Following the proof in [5,7], we define the state error vector by $e(t) = x(t) - \hat{x}(t)$. The time-derivative of $e(t)$ is obtained $\forall x, \hat{x} \in \Omega_{\hat{\rho}}$, $u \in U$ and $w(t) \in W$ as follows:

$$\begin{aligned} |\dot{e}| &= |F(x, u, w) - F_{nn}(\hat{x}, u)| \\ &\leq |F(x, u, w) - F(\hat{x}, u, 0)| + |F(\hat{x}, u, 0) - F_{nn}(\hat{x}, u)| \\ &\leq L_x|e(t)| + L_w w_m + \nu_m \end{aligned} \tag{21}$$

where the last inequality is derived from Equation (3b) and the fact that $|\nu| \leq \nu_m$. Since $x(t)$ and $\hat{x}(t)$ share the same initial condition, (i.e., $e(0) = 0$), the upper bound for $|e(t)|$ is derived for all $x(t), \hat{x}(t) \in \Omega_{\hat{\rho}}$ and $|w(t)| \leq w_m$ as follows:

$$|e(t)| = |x(t) - \hat{x}(t)| \leq \frac{L_w w_m + \nu_m}{L_x}(e^{L_x t} - 1) \tag{22}$$

Subsequently, $\forall x, \hat{x} \in \Omega_\rho$, Equation (20b) is derived based on the Taylor series expansion of $\hat{V}(x)$ around $\hat{x}$ as follows:

$$\begin{aligned} \hat{V}(x) &\leq \hat{V}(\hat{x}) + \frac{\partial \hat{V}(\hat{x})}{\partial x}|x-\hat{x}| + \kappa |x-\hat{x}|^2 \\ &\leq \hat{V}(\hat{x}) + \frac{\hat{c}_4\sqrt{\hat{\rho}}}{\sqrt{\hat{c}_1}}|x-\hat{x}| + \kappa |x-\hat{x}|^2 \end{aligned} \tag{23}$$

where κ is a positive real number, and the last inequality is derived using Equations (15a) and (15c). □

The above proposition demonstrates that the error of state trajectories of the nonlinear system of Equation (1) and the RNN model of Equation (4), starting from the same initial condition, is bounded for finite time. The next proposition is developed to demonstrate that if $x \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_s}$, the stabilizing controller $u = \Phi_{nn}(x) \in U$ implemented in a sample-and-hold fashion is able to maintain $\dot{\hat{V}}(x)$ negative such that the state will be driven towards the origin and ultimately enters a small neighborhood around the origin (i.e., Ω_{ρ_s}) in finite sampling steps.

Proposition 3. *Consider the system of Equation (1) under the controller $u = \Phi_{nn}(\hat{x}) \in U$ that meets the conditions of Equation (15) and is implemented in a sample-and-hold fashion, i.e., $u(t) = \Phi_{nn}(\hat{x}(t_k))$, $\forall t \in [t_k, t_{k+1})$, where $t_{k+1} := t_k + \Delta$. Let $\epsilon_w, \epsilon_s > 0$, $\Delta > 0$ and $\hat{\rho} > \hat{\rho}_e > \rho_s > 0$ satisfy*

$$-\frac{\hat{c}_3}{\hat{c}_2}\rho_s + L_{nn}M_{nn}\Delta \leq -\epsilon_s \tag{24a}$$

$$-\frac{\tilde{c}_3}{\hat{c}_2}\rho_s + L_x^{'}M\Delta + L_w^{'}w_m \leq -\epsilon_w \tag{24b}$$

$$\hat{\rho}_e > \max\{\hat{V}(\hat{x}(t_k+\Delta)) \mid \hat{x}(t_k) \in \Omega_{\rho_s}, u \in U, w \in W\} \tag{24c}$$

Then, for any $x(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_s}$, the following inequalities holds:

$$\hat{V}(\hat{x}(t)) \leq \hat{V}(\hat{x}(t_k)),\ \forall t \in [t_k, t_{k+1}) \tag{25a}$$

$$\hat{V}(x(t)) \leq \hat{V}(x(t_k)),\ \forall t \in [t_k, t_{k+1}) \tag{25b}$$

Proof. To show that the value of $\hat{V}$ decreases along the trajectory $\hat{x}(t)$ of the RNN model of Equation (4) over $t \in [t_k, t_{k+1})$, we calculate the time-derivative of $\hat{V}(\hat{x})$ based on $\hat{x}(t)$ as follows:

$$\begin{aligned} \dot{\hat{V}}(\hat{x}(t)) =& \frac{\partial \hat{V}(\hat{x}(t))}{\partial \hat{x}} F_{nn}(\hat{x}(t), \Phi_{nn}(\hat{x}(t_k))) \\ =& \frac{\partial \hat{V}(\hat{x}(t_k))}{\partial \hat{x}} F_{nn}(\hat{x}(t_k), \Phi_{nn}(\hat{x}(t_k))) \\ &+ \frac{\partial \hat{V}(\hat{x}(t))}{\partial \hat{x}} F_{nn}(\hat{x}(t), \Phi_{nn}(\hat{x}(t_k))) \\ &- \frac{\partial \hat{V}(\hat{x}(t_k))}{\partial \hat{x}} F_{nn}(\hat{x}(t_k), \Phi_{nn}(\hat{x}(t_k))) \end{aligned} \tag{26}$$

Using Equations (15a) and (15b) and the Lipschitz condition of Equation (16), the following inequalities are obtained:

$$\begin{aligned} \dot{\hat{V}}(\hat{x}(t)) \leq & -\frac{\hat{c}_3}{\hat{c}_2}\rho_s + \frac{\partial \hat{V}(\hat{x}(t))}{\partial \hat{x}} F_{nn}(\hat{x}(t), \Phi_{nn}(\hat{x}(t_k))) \\ & - \frac{\partial \hat{V}(\hat{x}(t_k))}{\partial \hat{x}} F_{nn}(\hat{x}(t_k), \Phi_{nn}(\hat{x}(t_k))) \\ \leq & -\frac{\hat{c}_3}{\hat{c}_2}\rho_s + L_{nn}M_{nn}\Delta \end{aligned} \tag{27}$$

Therefore, $\dot{\hat{V}}(\hat{x}(t)) \leq -\epsilon_s$ holds $\forall \hat{x}(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_s}$ and $t \in [t_k, t_{k+1})$ if Equation (24a) is satisfied. By integrating the above inequality, it follows that $\hat{V}(\hat{x}(t)) \leq \hat{V}(\hat{x}(t_k)) - \Delta\epsilon_s$, $\forall \hat{x}(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_s}$, $t \in [t_k, t_{k+1})$ (i.e., Equation (25a).

Next, to show that $\hat{V}(x(t)) \leq \hat{V}(x(t_k))$ holds for all $t \in [t_k, t_{k+1})$, the time-derivative of $\hat{V}(x)$ for the nonlinear system of Equation (1) (i.e., $\dot{x} = F(x, u, w)$) in the presence of bounded disturbances (i.e., $|w(t)| \leq w_m$) is derived as follows:

$$\begin{aligned}
\dot{\hat{V}}(x(t)) &= \frac{\partial \hat{V}(x(t))}{\partial x} F(x(t), \Phi_{nn}(x(t_k)), w) \\
&= \frac{\partial \hat{V}(x(t_k))}{\partial x} F(x(t_k), \Phi_{nn}(x(t_k)), 0) \\
&+ \frac{\partial \hat{V}(x(t))}{\partial x} F(x(t), \Phi_{nn}(x(t_k)), w) \\
&- \frac{\partial \hat{V}(x(t_k))}{\partial x} F(x(t_k), \Phi_{nn}(x(t_k)), 0)
\end{aligned} \tag{28}$$

Since Equation (18) shows that $\frac{\partial \hat{V}(x(t_k))}{\partial x} F(x(t_k), \Phi_{nn}(x(t_k)), 0) \leq -\tilde{c}_3 |x(t_k)|^2$ holds for all $x \in \Omega_{\hat{\rho}}$, the following inequality is obtained for all $x(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_s}$, $t \in [t_k, t_{k+1})$ using Equation (2a) and the Lipschitz condition in Equation (3):

$$\begin{aligned}
\dot{\hat{V}}(x(t)) &\leq -\frac{\tilde{c}_3}{\hat{c}_2} \rho_s + L'_x |x(t) - x(t_k)| + L'_w |w| \\
&\leq -\frac{\tilde{c}_3}{\hat{c}_2} \rho_s + L'_x M \Delta + L'_w w_m
\end{aligned} \tag{29}$$

Therefore, the following inequality is further derived for all $x(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_s}$ and $t \in [t_k, t_{k+1})$ if the condition of Equation (24b) is satisfied:

$$\dot{\hat{V}}(x(t)) \leq -\epsilon_w \tag{30}$$

Similarly, this implies that $\hat{V}(x(t)) \leq \hat{V}(x(t_k)) - \Delta\epsilon_w$, $\forall x(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_s}$, $t \in [t_k, t_{k+1})$. Therefore, the state of the nonlinear system of Equation (1) will enter Ω_{ρ_s} in finite sampling periods if $x(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_s}$. Additionally, if $x(t_k) \in \Omega_{\rho_s}$, in which Equations (29) and (30) do not hold, Equation (24c) guarantees that the state will not leave $\Omega_{\hat{\rho}_e}$ in one sampling period for all $u \in U$ and $w \in W$. If the state $x(t_{k+1})$ leaves Ω_{ρ_s} but remains in $\Omega_{\hat{\rho}_e}$, at the next sampling period $t \in [t_{k+1}, t_{k+2})$, Equation (30) is satisfied again and the state will be driven towards the origin. Therefore, the state of the nonlinear system of Equation (1) is bounded in $\Omega_{\hat{\rho}}$ for all times. □

Based on Propositions 2 and 3, the following theorem is established to show recursive feasibility of the LEMPC optimization problem of Equation (19), and the boundedness of the closed-loop state within $\Omega_{\hat{\rho}}$ under the sample-and-hold implementation of the LEMPC.

Theorem 2. *Consider the closed-loop system of Equation (1) under the sample-and-hold implementation of the LEMPC of Equation (19) with the stabilizing controller $\Phi_{nn}(x)$ that satisfies Equation (15). Let $\Delta > 0$, $\epsilon_w > 0$ and $\hat{\rho} > \hat{\rho}_e > 0$ satisfy Equation (24) and the following inequality:*

$$\hat{\rho}_e \leq \hat{\rho} - \frac{\hat{c}_4 \sqrt{\hat{\rho}}}{\sqrt{\hat{c}_1}} f_w(\Delta) - \kappa (f_w(\Delta))^2 \tag{31}$$

If $x_0 \in \Omega_{\hat{\rho}}$ and the conditions of Propositions 2 and 3 are satisfied, then there always exists a feasible solution for the optimization problem of Equation (19), and the closed-loop state $x(t)$ is bounded in the closed-loop stability region $\Omega_{\hat{\rho}}$, $\forall t \geq 0$.

Proof. The LEMPC of Equation (19) predicts future states by averaging the results of an ensemble of RNN models in Equation (19b). Since each RNN model is trained to satisfy the modeling error constraint and Equation (16), it is readily shown that the averaged results of an ensemble of multiple RNN models also satisfy the above conditions, and thus, the results derived in Propositions 1–3 for a single RNN model can be generalized to an RNN ensemble.

We first prove that the optimization problem of Equation (19) can be solved with recursive feasibility for all $x \in \Omega_{\hat{\rho}}$. Specifically, if $x(t_k) \in \Omega_{\hat{\rho}_e}$, the control actions $\Phi_{nn}(x(t_{k+i}))$, $i = 0, 1, ..., N-1$ satisfy the input constraint of Equation (19c) and the Lyapunov-based constraint of Equation (19e) since it is shown in Equation (25a) that the states predicted by the RNN model of Equation (19b) remain inside $\Omega_{\hat{\rho}_e}$ under the controller $\Phi_{nn}(x)$. Additionally, if $x(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\hat{\rho}_e}$, the control action $u(t) = \Phi_{nn}(x(t_k)) \in U$, $t \in [t_k, t_{k+1})$ satisfies the input constraint of Equation (19c) and the Lyapunov-based constraint of Equation (19f) such that the state can be driven towards the origin during the next sampling period. Therefore, the stabilizing controller $u = \Phi_{nn}(x) \in U$ provides a feasible solution that satisfies all the constraints of the LEMPC optimization problem of Equation (19) if $x(t) \in \Omega_{\hat{\rho}}$ for all times.

Next, we prove that for $x_0 \in \Omega_{\hat{\rho}}$, the state of the closed-loop system of Equation (1) is bounded in $\Omega_{\hat{\rho}}$ for all times. Specifically, if $x(t_k) \in \Omega_{\hat{\rho}_e}$, the predicted states $\hat{x}(t)$ of the RNN model of Equation (19b) are maintained in $\Omega_{\hat{\rho}_e}$ under the constraint of Equation (19e). According to Proposition 2, the actual state $x(t)$, $t \in [t_k, t_{k+1})$ of the nonlinear system of Equation (1) is bounded by the following inequality:

$$\begin{aligned}\hat{V}(x) &\leq \hat{V}(\hat{x}) + \frac{\hat{c}_4\sqrt{\hat{\rho}}}{\sqrt{\hat{c}_1}}|x-\hat{x}| + \kappa|x-\hat{x}|^2 \\ &\leq \hat{V}(\hat{x}) + \frac{\hat{c}_4\sqrt{\hat{\rho}}}{\sqrt{\hat{c}_1}}f_w(\Delta) + \kappa(f_w(\Delta))^2\end{aligned} \tag{32}$$

Therefore, if $\Omega_{\hat{\rho}_e}$ is chosen as a level set of $\hat{V}$ that satisfies Equation (31), $\hat{V}(x)$ based on the actual state $x(t)$ is bounded in $\Omega_{\hat{\rho}}$ for all $t \in [t_k, t_{k+1})$. However, if $x(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\hat{\rho}_e}$, the constraint of Equation (19f) is activated such that the control action u decreases the value of $\hat{V}(\hat{x})$ based on the states predicted by the RNN model of Equation (19b) within the next sampling period. According to Equation (25b) in Proposition 3, the value of $\hat{V}$ also decreases along the state trajectory of the actual nonlinear system of Equation (1) over $t \in [t_k, t_{k+1})$. Therefore, it is concluded that for any initial condition in $\Omega_{\hat{\rho}}$, the closed-loop state of the system of Equation (1) is bounded in $\Omega_{\hat{\rho}}$ for all times under the LEMPC of Equation (19). □

5. Application to a Chemical Process Example

A chemical process example is used to illustrate the application of LEMPC using an ensemble of RNN models, under which the state of the closed-loop system is maintained within the stability region in state-space for all times. We consider a well-mixed, non-isothermal continuous stirred tank reactor (CSTR), in which an irreversible second-order exothermic reaction takes place. Specifically, the reaction transforms a reactant A to a product B ($A \rightarrow B$). The inlet temperature, the inlet concentration of A, and the feed volumetric flow rate of the reactor are T_0, C_{A0} and F, respectively. Additionally, the CSTR is equipped with a heating jacket to supply or remove heat at a rate Q. The following material and energy balance equations are used to represent the CSTR dynamic model:

$$\frac{dC_A}{dt} = \frac{F}{V}(C_{A0} - C_A) - k_0 e^{\frac{-E}{RT}} C_A^2 \tag{33a}$$

$$\frac{dT}{dt} = \frac{F}{V}(T_0 - T) + \frac{-\Delta H}{\rho_L C_p} k_0 e^{\frac{-E}{RT}} C_A^2 + \frac{Q}{\rho_L C_p V} \tag{33b}$$

where V is the volume of the reacting liquid in the reactor, C_A is the reactant A concentration in the reactor, Q is the heat input rate and T is the temperature of the reactor. The feed volumetric flow rate and temperature are F and T_0, respectively. The feed concentration of reactant A is C_{A0}. The reacting liquid has a constant heat capacity of C_p and density of ρ_L. k_0 is the pre-exponential constant, ΔH is the enthalpy of reaction, R is the ideal gas constant, and E is the activation energy. Process parameter values are listed in Table 1.

Table 1. Parameter values of the CSTR.

$T_0 = 300$ K	$F = 5$ m^3/h
$V = 1$ m^3	$E = 5 \times 10^4$ kJ/kmol
$k_0 = 8.46 \times 10^6$ m^3/kmol h	$\Delta H = -1.15 \times 10^4$ kJ/kmol
$C_p = 0.231$ kJ/kg K	$R = 8.314$ kJ/kmol K
$\rho_L = 1000$ kg/m^3	$C_{A0_s} = 4$ kmol/m^3
$Q_s = 0.0$ kJ/h	$C_{A_s} = 1.2$ kmol/m^3
$T_s = 438$ K	

The CSTR is initially operated at the unstable steady-state $(C_{As}, T_s) = (1.95$ kmol/m^3, 402 K), and $(C_{A0_s}\ Q_s) = (4$ kmol/m^3, 0 kJ/h). In this example, the two manipulated inputs are the heat input rate, $\Delta Q = Q - Q_s$, and the inlet concentration of species A, $\Delta C_{A0} = C_{A0} - C_{A0_s}$, respectively. Additionally, the manipulated inputs are subject to the following constraints: $|\Delta Q| \leq 5 \times 10^5$ kJ/h and $|\Delta C_{A0}| \leq 3.5$ kmol/m^3. The states and the inputs of the system of Equation (33) are represented by $x^T = [C_A - C_{As}\ T - T_s]$ and $u^T = [\Delta C_{A0}\ \Delta Q]$, respectively. Therefore, the unstable steady-state of the system of Equation (33) is at the origin of the state-space, (i.e., $(x_s^*, u_s^*) = (0, 0)$).

The control objective of the LEMPC of Equation (19) is to maximize the production rate of B in the CSTR by manipulating the inlet concentration ΔC_{A0} and the heat input rate ΔQ, while maintaining the closed-loop state trajectories in the stability region $\Omega_{\hat{\rho}}$ for all times under LEMPC. The objective function of the LEMPC optimizes the production rate of B as follows:

$$l_e(\tilde{x}, u) = k_0 e^{-E/RT} C_A^2 \tag{34}$$

To simulate the dynamic model of Equation (33) numerically, the explicit Euler method is used with an integration time step of $h_c = 10^{-4}$ h. The Python module of the IPOPT software package [27], named PyIpopt, is used to solve the nonlinear optimization problem of the LEMPC of Equation (19) with a sampling period $\Delta = 10^{-2}$ h. Additionally, a Message Passing Interface (MPI) for the Python programming language, named *MPI4Py* ([28]), is incorporated in the LEMPC optimization problem to execute multiple RNN predictions of Equation (19b) concurrently in independent computing processes.

Extensive open-loop simulations for the CSTR system of Equation (33) are initially conducted to generate the dataset for RNN models. The control Lyapunov function $\hat{V}(x) = x^T P x$ is designed with the following positive definite P matrix:

$$P = \begin{bmatrix} 1060 & 22 \\ 22 & 0.52 \end{bmatrix} \tag{35}$$

Two hidden recurrent layers consisting of 96 and 64 recurrent units, respectively, are used in the RNN model. Additionally, a 10-fold cross validation are used to construct an ensemble of RNN models for the LEMPC of Equation (19). The closed-loop stability region $\Omega_{\hat{\rho}}$ with $\hat{\rho} = 368$ and a subset $\Omega_{\hat{\rho}_e}$ with $\hat{\rho}_e = 320$ for the CSTR system are characterized based on the obtained RNN models and the stabilizing controller $u = \Phi_{nn}(x) \in U$. To demonstrate the effectiveness of the proposed LEMPC of Equation (19), a linear state-space model (i.e., $\dot{x} = Ax + Bu$ with $A = 100 \times [-0.154\ \ -0.003;\ 5.19\ \ 0.138]$ and $B = [4.03\ \ 0;\ 1.23\ \ 0.004]$) is also identified following the system identification method in [29], and is added into the following performance comparison under the ensemble of RNN models and the first-principles model of Equation (33).

The open-loop simulation studies are first carried out for the first-principles model of Equation (33), an RNN model ensemble, and the linear state-space model, respectively. The optimal number of RNN models used in the ensemble is determined to be five from extensive simulation tests due to its best performance in terms of closeness to the trajectories under the first-principles model. As shown in Figure 5, starting from various initial conditions in $\Omega_{\hat{\rho}}$, the state trajectories under the RNN model ensemble almost overlap those under the first-principles model using the same input sequences, while the state trajectories under the linear state-space model show considerable deviation, especially in the top and bottom regions of $\Omega_{\hat{\rho}}$. From Figure 5, it is demonstrated that the RNN model ensemble is a good representation for the first-principles model of Equation (33) in that it successfully captures the CSTR nonlinear dynamics everywhere in $\Omega_{\hat{\rho}}$.

Subsequently, the closed-loop simulation of the nominal CSTR system (i.e., $w(t) \equiv 0$) under the first-principles model of Equation (33), the ensemble of RNN models, and the linear state-space model, respectively, are shown in Figure 6. The following material constraint is used in the LEMPC of Equation (19) to make the averaged reactant material available within the entire operating period t_p to be its steady-state value, C_{A0s} (i.e., the averaged reactant material in deviation form, u_1, is equal to 0).

$$\frac{1}{t_p}\int_0^{t_p} u_1(\tau)d\tau = 0\ \text{kmol/m}^3 \tag{36}$$

In Figure 6, it is shown that for the initial condition (0,0), the state trajectory under the ensemble of RNN models is close to that under the first-principles model, and both state trajectories are bounded in $\Omega_{\hat{\rho}_e}$ in the absence of disturbances. However, it is observed that the state trajectory under the linear state-space model ultimately leaves $\Omega_{\hat{\rho}}$ due to its large model mismatch. This implies that a more conservative set $\Omega_{\hat{\rho}_e}$ needs to be characterized for the state spaced model to guarantee the boundedness of state in the stability region $\Omega_{\hat{\rho}}$.

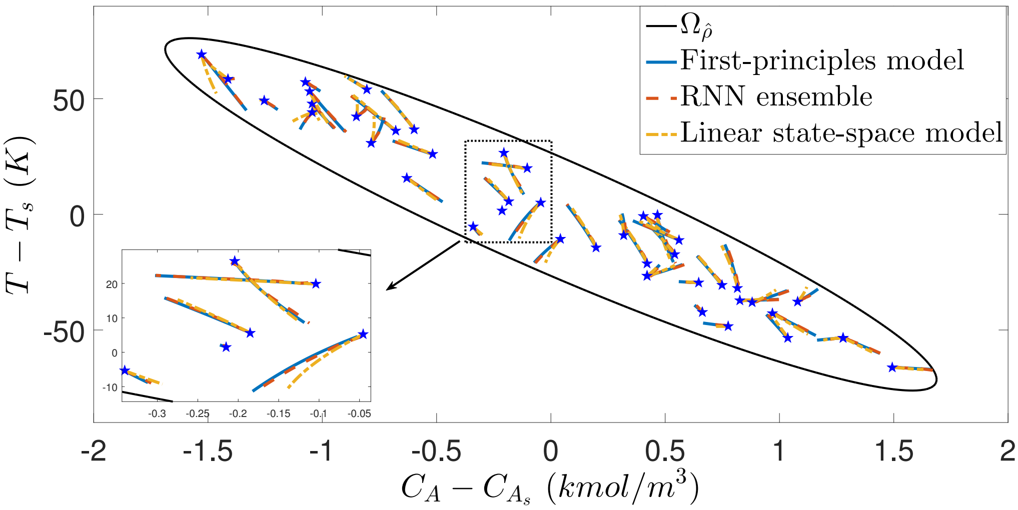

Figure 5. The state trajectories for the open-loop simulation using the first-principles model of Equation (33), the ensemble of RNN models, and the linear state-space model, respectively for various sets of inputs and initial conditions (marked as blue stars) x_0 in the closed-loop stability region $\Omega_{\hat{\rho}}$.

Additionally, Figures 7 and 8 show the manipulated inputs profiles in deviation from the steady-state values. It is shown that the manipulated inputs are bounded within the input constraints for all times. Moreover, the material constraint of Equation (36) is satisfied as it is shown in Figure 7 that the LEMPC using the first-principles model and the RNN model ensemble both consume the maximum allowable ΔC_{A0} at the first few sampling steps, and thus have to lower the consumption of ΔC_{A0} near the end of operating period. The LEMPC using the linear state-space model also satisfies the material constraint, but shows persistent oscillation due to model mismatch. Additionally, in Figure 8, it is shown that the consumption of ΔQ under the RNN ensemble is close to that under the first-principles model (both correspond to left y-axis) since their closed-loop trajectories shown in Figure 6 are similar. However, ΔQ shows large oscillation under the LEMPC using the linear state-space model (right y-axis) since the closed-loop trajectory in Figure 6 leaves $\Omega_{\hat{\rho}_e}$, and thus, the contractive constraint of Equation (19f) is activated frequently.

To demonstrate that the closed-loop system under LEMPC achieves high process economics than the steady-state operation using the same amount reactant C_{A0}, we calculate the accumulated economic benefits $L_E = \int_0^{t_p} l_e(x,u)dt$ over the entire operating period t_p = 0.2 h, which are 2.04, 4.14, and 4.22 for the steady-state operation, the LEMPC under an ensemble of RNN models, and the LEMPC under the first-principles model, respectively. Therefore, it is demonstrated that both closed-loop stability and economic optimality are achieved under the proposed LEMPC using an ensemble of RNN models.

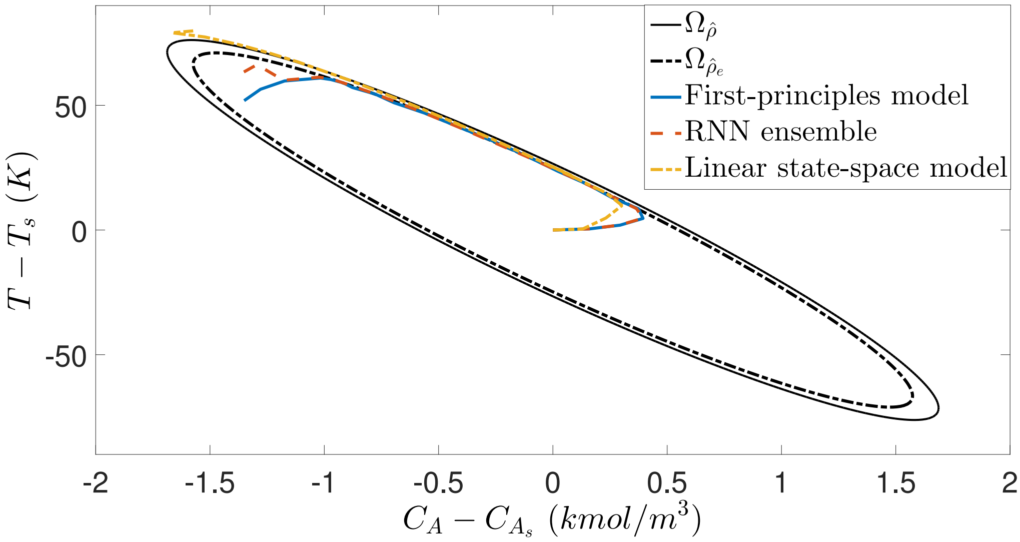

Figure 6. The state-space profiles for the closed-loop CSTR under the LEMPC using the following models: the first-principles model (blue trajectory), the RNN model ensemble (red trajectory) and the linear state-space model (yellow trajectory) for an initial condition (0, 0).

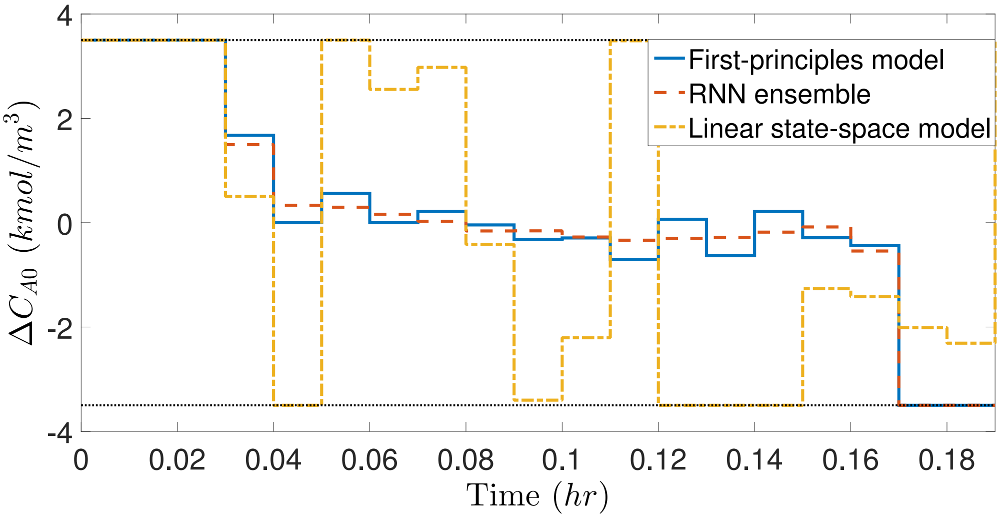

Figure 7. Manipulated input profiles ($u_1 = \Delta C_{A0}$) for the initial condition (0, 0) under the LEMPC using the following models: the first-principles model (blue trajectory), the RNN model ensemble (red trajectory) and the linear state-space model (yellow trajectory), where the dashed black horizontal lines represent the upper and lower bounds for ΔC_{A0}.

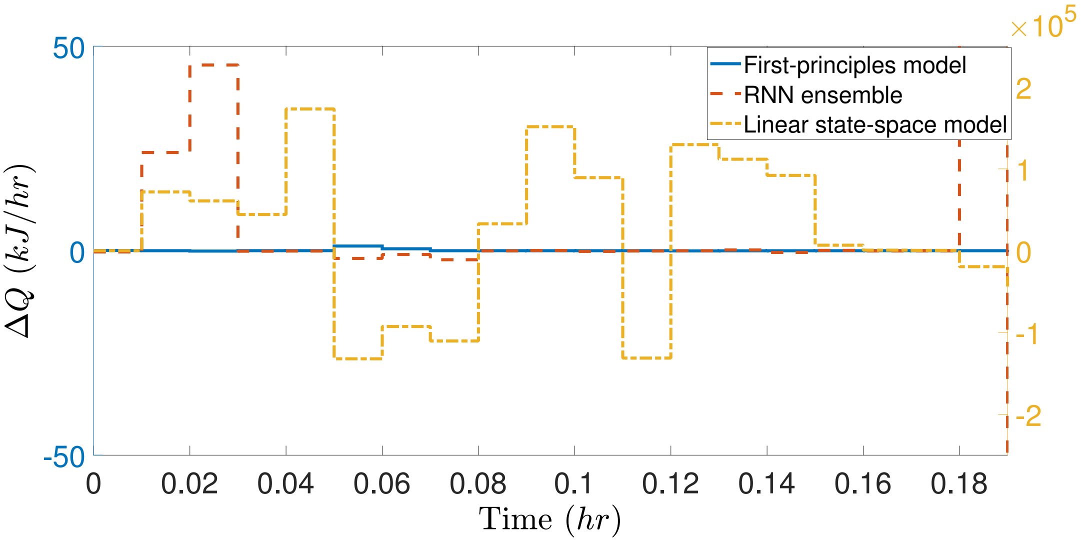

Figure 8. Manipulated input profiles ($u_2 = \Delta Q$) for the initial condition (0, 0) under the LEMPC using the following models: the first-principles model (blue trajectory corresponding to left y-axis), the RNN model ensemble (red trajectory corresponding to left y-axis) and the linear state-space model (yellow trajectory corresponding to right $y - axis$).

6. Conclusions

This work presented a new class of economic model predictive controllers that use an ensemble of recurrent neural network models as the prediction model for nonlinear systems. Specifically, an RNN ensemble was first developed based on the dataset of extensive open-loop simulations within the operating region. Parallel computing was used to reduce the computation time of prediction by multiple RNN models. Then, the LEMPC that incorporates RNN models and Lyapunov-based stability constraints was formulated to stabilize a nonlinear process within the closed-loop stability region while optimizing process economic benefits simultaneously. The proposed LEMPC using an RNN ensemble was applied to a nonlinear chemical process example to demonstrate its economic optimality and closed-loop stability.

Author Contributions: Z.W. developed the idea of machine learning-based EMPC for nonlinear systems, performed the simulation studies and prepared the initial draft of the paper. P.D.C. oversaw all aspects of the research and revised this manuscript.

Funding: Financial support from the National Science Foundation and the Department of Energy is gratefully acknowledged.

Conflicts of Interest: The authors declare that they have no conflict of interest regarding the publication of the research article.

References

1. Angeli, D.; Amrit, R.; Rawlings, J.B. On average performance and stability of economic model predictive control. *IEEE Trans. Autom. Control* **2012**, *57*, 1615–1626. [CrossRef]
2. Müller, M.A.; Angeli, D.; Allgöwer, F. Economic model predictive control with self-tuning terminal cost. *Eur. J. Control* **2013**, *19*, 408–416. [CrossRef]
3. Ellis, M.; Durand, H.; Christofides, P.D. A tutorial review of economic model predictive control methods. *J. Process Control* **2014**, *24*, 1156–1178. [CrossRef]
4. Heidarinejad, M.; Liu, J.; Christofides, P.D. Economic model predictive control of nonlinear process systems using Lyapunov techniques. *AIChE J.* **2012**, *58*, 855–870. [CrossRef]
5. Wu, Z.; Durand, H.; Christofides, P.D. Safe economic model predictive control of nonlinear systems. *Syst. Control Lett.* **2018**, *118*, 69–76. [CrossRef]

6. Wu, Z.; Zhang, J.; Zhang, Z.; Albalawi, F.; Durand, H.; Mahmood, M.; Mhaskar, P.; Christofides, P.D. Economic model predictive control of stochastic nonlinear systems. *AIChE J.* **2018**, *64*, 3312–3322. [CrossRef]
7. Alanqar, A.; Ellis, M.; Christofides, P.D. Economic model predictive control of nonlinear process systems using empirical models. *AIChE J.* **2015**, *61*, 816–830. [CrossRef]
8. Alanqar, A.; Durand, H.; Christofides, P.D. On identification of well-conditioned nonlinear systems: Application to economic model predictive control of nonlinear processes. *AIChE J.* **2015**, *61*, 3353–3373. [CrossRef]
9. Wang, T.; Gao, H.; Qiu, J. A combined adaptive neural network and nonlinear model predictive control for multirate networked industrial process control. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, *27*, 416–425. [CrossRef]
10. Wu, Z.; Tran, A.; Ren, Y.M.; Barnes, C.S.; Chen, S.; Christofides, P.D. Model Predictive Control of Phthalic Anhydride Synthesis in a Fixed-Bed Catalytic Reactor via Machine Learning Modeling. *Chem. Eng. Res. Des.* **2019**, *145*, 173–183. [CrossRef]
11. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.S.; Davis, A.; Dean, J.; Devin, M.; et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. *arXiv* **2015**, arXiv:1603.04467.
12. Jia, Y.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J.; Girshick, R.; Guadarrama, S.; Darrell, T. Caffe: Convolutional Architecture for Fast Feature Embedding. *arXiv* **2014**, arXiv:1408.5093.
13. Chollet, F. Keras. 2015. Available online: https://keras.io (accessed on 10 March 2019).
14. Tu, J.V. Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes. *J. Clin. Epidemiol.* **1996**, *49*, 1225–1231. [CrossRef]
15. Hopfield, J.J. Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci. USA* **1982**, *79*, 2554–2558. [CrossRef] [PubMed]
16. Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Netw.* **2015**, *61*, 85–117. [CrossRef] [PubMed]
17. Venkatasubramanian, V. The promise of artificial intelligence in chemical engineering: Is it here, finally? *AIChE J.* **2019**, *65*, 466–478. [CrossRef]
18. Sewell, M. *Ensemble Learning*; Technical report; UCL: London, UK, 2010.
19. Zhang, C.; Ma, Y. *Ensemble Machine Learning: Methods and Applications*; Springer: Berlin, Germany, 2012.
20. Almasi, G.S.; Gottlieb, A. *Highly Parallel Computing*; Benjamin-Cummings Publishing Co., Inc.: Redwood City, CA, USA, 1988.
21. Lin, Y.; Sontag, E.D. A universal formula for stabilization with bounded controls. *Syst. Control Lett.* **1991**, *16*, 393–397. [CrossRef]
22. Sontag, E.D. Neural nets as systems models and controllers. In Proceedings of the Seventh Yale Workshop on Adaptive and Learning Systems, New Haven, CT, USA, 20–22 May 1992; pp. 73–79.
23. Kosmatopoulos, E.B.; Polycarpou, M.M.; Christodoulou, M.A.; Ioannou, P.A. High-order neural network structures for identification of dynamical systems. *IEEE Trans. Neural Netw.* **1995**, *6*, 422–431. [CrossRef]
24. Dua, D.; Graff, C. UCI Machine Learning Repository. 2017. Available online: http://archive.ics.uci.edu/ml. (accessed on 10 March 2019).
25. Sayyar-Rodsari, B.; Hartman, E.; Plumer, E.; Liano, K.; Schweiger, C. Extrapolating gain-constrained neural networks-effective modeling for nonlinear control. In Proceedings of the 43rd IEEE Conference on Decision and Control, Nassau, Bahamas, 14–17 December 2004; Volume 5, pp. 4964–4971.
26. Mendes-Moreira, J.; Soares, C.; Jorge, A.M.; Sousa, J.F.D. Ensemble approaches for regression: A survey. *ACM Comput. Surv. (CSUR)* **2012**, *45*, 10. [CrossRef]
27. Wächter, A.; Biegler, L.T. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Math. Program.* **2006**, *106*, 25–57. [CrossRef]
28. Dalcin, L.D.; Paz, R.R.; Kler, P.A.; Cosimo, A. Parallel distributed computing using Python. *Adv. Water Resour.* **2011**, *34*, 1124–1139. [CrossRef]
29. Kheradmandi, M.; Mhaskar, P. Data Driven Economic Model Predictive Control. *Mathematics* **2018**, *6*, 51. [CrossRef]