

Article

RNN-LSTM-Based Model Predictive Control for a Corn-to-Sugar Process

Jiaqi Meng¹, Chengbo Li¹, Jin Tao², Yi Li², Yi Tong², Yu Wang², Lei Zhang¹, Yachao Dong^{1,*}
and Jian Du^{1,*}

¹ Institute of Chemical Process Systems Engineering, School of Chemical Engineering, Dalian University of Technology, Dalian 116024, China

² COFCO Biotechnology Co., Ltd., Beijing 100005, China

* Correspondence: yachao.dong@dlut.edu.cn (Y.D.); dujian@dlut.edu.cn (J.D.)

Abstract: The corn-to-sugar process is difficult to control automatically because of the complex physical and chemical phenomena involved. Because the RNN-LSTM model has been shown to handle long-term time dependencies well, this article focused on the design of a model predictive control system based on this machine learning model. Based on the historical data, we first reduced the input variable dimension through data preprocessing, data dimension reduction, sensitivity analysis, etc., and then the RNN-LSTM model, with these identified key sites as inputs, and the dextrose equivalent value as the output, was constructed. Then, through model predictive control using the locally linearized RNN-LSTM as the predictive model, the objective value of the dextrose equivalent was successfully controlled at the target value by our simulation study, in different situations of setpoint changes and disturbances. This showed the potential of applying RNN-LSTM-Based model predictive control in a corn-to-sugar process.

Keywords: corn-to-sugar process; RNN-LSTM; model predictive control; data-driven method



Citation: Meng, J.; Li, C.; Tao, J.; Li, Y.; Tong, Y.; Wang, Y.; Zhang, L.; Dong, Y.; Du, J. RNN-LSTM-Based Model Predictive Control for a Corn-to-Sugar Process. *Processes* **2023**, *11*, 1080. <https://doi.org/10.3390/pr11041080>

Academic Editors: Zhe Wu, Xunyuan Yin, Yan Qin and Xiaodong Xu

Received: 28 February 2023

Revised: 31 March 2023

Accepted: 1 April 2023

Published: 3 April 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Corn (*Zea Mays*) is one of the three major food crops in the world, and it is also the main raw material for concentrated feed. Compared with sweet potatoes, corn is an important food crop and raw material for industrial production in the world, due to its rich starch, protein, and fat content [1]. The corn-sugar production process includes many production sections, such as the starch section, fructose section, corn-feeding section, starch detection, fructose detection, citric-acid section, etc. The process is complicated, and the generated data have a complex structure, leading to the need for a lot of resources to control the key nodes of the corn plant. Therefore, model predictive control (MPC) is widely used in corn factories, because it is a very effective control strategy for multi-input and multi-output systems.

An MPC process can usually be conceptually described with the following steps. At each moment, an optimization problem is first solved online, based on the current measurement information obtained, and the first element of the calculated control sequence is applied to the controlled object. At the next sampling moment, the above process is repeated: the new measured value is used as the initial condition to predict the future dynamics of the system at this time, and the optimization problem is refreshed and solved again. Since the MPC concept was first proposed, MPC has been used in many fields, such as power systems [2], vehicles [3,4] and rail transportation [5]. Hrovat et al. [6] found that MPC was widely used in the automotive industry and shows better closed-loop performance compared to traditional control solutions. In chemical engineering, Mendis and Wickramasinghe et al. [7] applied MPC to maximize the product purity of a batch distillation tower and studied its performance in Matlab. Prasad et al. [8] controlled the

filled height of a conical-shaped tank, identifying three separate linear models at different heights to design one controller for each and combined the outputs as an ensemble to obtain a general output for the manipulated variable. In all these processes in which model predictive control is applied, mathematical models describing the relationship between manipulated inputs and process outputs are essential for building model-based control systems for industrial applications [9].

However, due to the complexity of the physical and chemical phenomena in a chemical factory, first-principle models are often difficult to derive. Therefore, with the development of computer hardware and artificial intelligence technology, model predictive control based on neural network methods has gradually received more and more attention, due to its excellent ability to represent complex physical models. Among the many machine learning techniques, the recurrent neural network (RNN) is most suitable for dealing with the ordered form of the time series datasets. The basic RNN model can handle certain short-term time dependencies, while the long short-term memory (LSTM) model based on RNN handles long-term time dependencies with improved performance and is widely used in nonlinear model systems [10]. RNN-LSTM is a variant of RNN, with the concepts of the cell state and the gate structure introduced. A sequence of cell states makes up the path for information transmission in the time sequence [11,12] and the gate structure will learn to save and forget the information during the training process. LSTM architecture was first introduced by Hochreiter and Schmidhuber in 1997 [13], and effectively solved the problems of gradient disappearance and gradient explosion that RNN is prone to. Afterwards, Alex Graves and Jurgen Schmidhuber [14] proposed a bidirectional long short-term memory (BLSTM) neural network based on the previous work, which was frequently used in the 2000s. In recent years, LSTM has been applied in many fields and more and more widely in engineering. Akhter and Mekhilef et al. [15] proposed an RNN-LSTM based on a simulated annealing algorithm (SSA) model for predicting the output power of photovoltaic systems, and proved that the model has better robustness. Demidova [16] presented the LSTM and gate recurrent unit (GRU) models for aircraft engine-maintenance problems and proved the superiority of LSTM in solving maintenance performance problems based on multidimensional time series.

In recent years, model predictive control based on neural networks has developed rapidly. Xu et al. [17] solved the four-tank benchmark problem based on an RNN model, and the accuracy of the model and the effectiveness of the proposed method were verified by experiments. Norouzi et al. [18] modeled the compression-ignition engine based on machine learning and used it to realize the model prediction controller for the 4.5-L Cummins diesel engine. By comparing it with the standard feedforward production controller, they found that the online optimized MPC scheme could reduce emissions of polluting gases, reduce fuel consumption, and reduce costs. Due to the inaccuracy of the prediction of domestic hot water in a single household or small system, Maltais and Gosselin [19] considered the MPC predicted by the machine learning model and compared it with the rule-based controller in order to reduce heat loss and verify the applicability of the MPC of the machine learning prediction model. The results showed that the MPC based on the machine learning prediction model could save about 8% of energy consumption. Wu et al. [20–22] successfully adjusted the product yield of high exothermic reactions in industrial-scale fixed-bed reactors by using MPC based on machine learning and introduced different integrated-learning methods in order to improve the model prediction performance of neural network models in batch polymerization reactors. Their research team followed up with more extensive and interesting research in the field of chemical engineering [23,24]. Zarzycki and Lawrynczuk [25] considered two simulated industrial processes: polymerization reactor and neutralization process and compared the efficiency of LSTM and GRU models under multiple model configurations. Wang et al. [26] proposed an LSTM and MPC control strategy weighted by particle swarm optimization to control the temperature of superheated steam and proved the advantages of LSTM-MPC over traditional proportion integration differentiation (PID) control and single MPC control

through simulation experiments. Singh et al. [27] established a three-layer feed-forward neural network (FNN) model for a binary continuous distillation column and used the MPC algorithm to predict the product purity to search for the optimal control moves. Compared with the traditional PID controller, the MPC scheme shows an improvement in the settling time. Qing et al. [28] construct MPC formulations for the distributed parameter systems (DPS) based on the learned, dimensional-reduced model. They used a path-integral optimal control algorithm for MPC implementation to avoid any analytic derivatives of the dynamics. The effectiveness of integration of a deep neural network-based model with MPC was demonstrated in a tubular reactor with recycle cases.

Although machine learning-based MPCs have had many applications in chemical engineering and is often used with specific equipment, such as a single reactor or a separation unit, few works focus on a complex process in a factory-section level. However, the process from corn-to-sugar is complex, with many factors that affect product quality. The traditional corn-to-sugar mill is susceptible to the influence of different corn varieties, which leads to fluctuations in product quality. It is difficult to realize the accurate adjustment of process parameters and maintain stable operation through engineering experience. Therefore, it is difficult to regulate this process by a mechanism or through experience alone. It is necessary to use big data technology to find the key parameters of the process, establish the process prediction model, and then carry out advanced control.

In this work, an RNN-LSTM model was established and an MPC based on the RNN-LSTM was applied to the corn-to-sugar process. In the second section, steps of the approach, including data preprocessing, dimensionality reduction methods, and RNN-LSTM based modeling, were described. In addition, an extended weights connection method was proposed for the RNN-LSTM model to analyze the importance of variables. In the third section, based on the established RNN-LSTM model, the key sites in the corn-to-sugar factory were analyzed, and the MPC was carried out, which was used to control the key sites in the corn-to-sugar factory to adjust the objective value. The modeling process and predictive control were implemented on the Matlab platform, details of which can be seen in the supporting information. Based on the successful simulation of this work, we will further test this neural network-based control strategy in the real plant.

2. RNN-LSTM-Based Model Predictive Control Framework

The framework of RNN-LSTM-based model predictive control includes two parts. Since the corn production factory has very complex physical and chemical processes, it is very difficult to establish a first-principles model, so the first part of the framework is to establish a black box model for the corn-to-sugar process. There are many commonly used neural network models, such as the feedforward neural network (FNN), convolutional neural network (CNN), RNN, LSTM, etc. Due to the strong temporal nature of factory data, RNN is widely used for modeling factory equipment. However, because the RNN model is affected by short-term memory and faces the problem of gradient disappearance, we used a variant of RNN, the RNN-LSTM model, to solve the problem of gradient disappearance to a certain extent. A comparison of CNN, ANN and LSTM is shown in Section 3.2.

The second part of the framework is to establish the MPC of the saccharification section based on the RNN-LSTM model. The completed RNN-LSTM model can predict the real-time output of future controlled variables based on the relevant state information provided by the sensors in the saccharate section and the input of future manipulated variables. By adopting the rolling finite time-domain optimization strategy, the MPC can solve the optimal control sequence of the finite period of time at each sampling time according to the optimization performance index of the time. The control of the saccharification section is realized by executing the input of the current time of the optimal control sequence. At the next sampling time, the MPC will follow the same steps to obtain the optimal control input to control the saccharification section.

2.1. Data Preprocessing

Data are the core of process modeling in the proposed approach. However, due to some realistic issues, the data collected from the factory is usually incomplete, noise-containing, inconsistent, and prone to other problems. Therefore, it is necessary to carry out a series of processing works, such as cleaning, integration, conversion, discretization, and specification of the original data to improve the efficiency of data analysis and the reliability of the results. According to the data characteristics collected from the factory, the data preprocessing steps in this paper are shown in Figure 1. Data preprocessing includes three parts. First, the data are cleaned, and the missing values and outliers are processed. Then, the initial dimension reduction of the data is carried out, and the 100 dimensions of the data are saved. Finally, the sensitivity analysis of the data after dimensionality reduction is carried out, and the 20-dimensional data are retained.

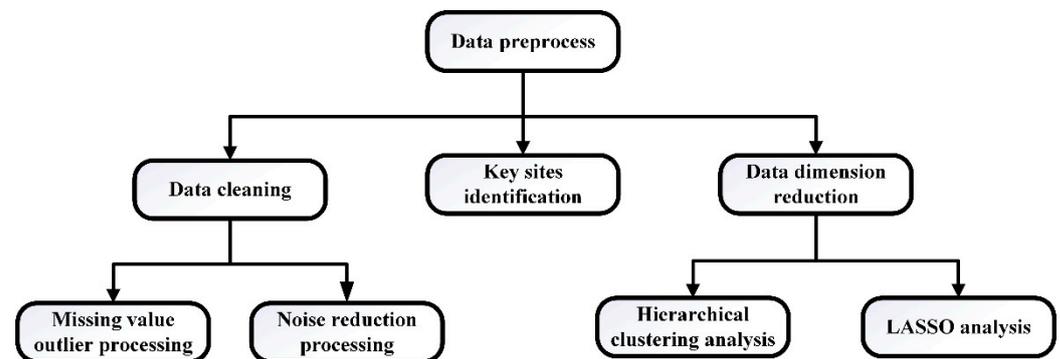


Figure 1. Preprocessing frame for factory data.

2.1.1. Data Cleaning

The purpose of data cleaning is to supplement the missing part, correct or delete the incorrect part, screen, and remove the redundant part, and, finally, organize it into high-quality data that are easy to be analyzed and used. The data cleaning methods considered in this paper mainly include missing value and outlier value processing, and data noise reduction.

1. Missing value and outlier processing

There are various reasons for missing values in factories, and these can mainly be divided into mechanical reasons and human reasons. A mechanical reason can be due to a series of problems such as on-site detection failure, plant shutdown and maintenance, while a human reason can be due to data loss caused by human subjective mistakes, historical limitations, or intentional concealment. There are two main solutions to deal with missing values: (a) take the average value in the case of a small number of missing values in a continuous period, and supplement according to the data before and after the period [29]; (b) when a large number of values are missing for a consecutive period of time, all data in the corresponding period are deleted [30]. Although outliers are rare, they will greatly affect the training efficiency and accuracy of the model. Therefore, we used the Pauta criterion [31] or Chauvenet method [32] to deal with outliers according to data characteristics.

2. Noise reduction processing

Due to the presence of interference data in the dataset provided by the factory, we adopted the $(2n + 1)$ simple moving average [33] method to smooth and filter the data noise. In this method, the n data $(y_i - n, \dots, y_i - 1, \dots, y_i + n)$ before and after the centre of y_i to obtain the average y'_i instead of y_i are determined, which can be expressed by Equation (1).

$$y'_i = \frac{1}{2n + 1} \sum_{k=-n}^n y_{i+k} \quad (1)$$

2.1.2. Data Dimension Reduction

The data provided in the factory are often multidimensional and there is a strong correlation between different data. Therefore, it is necessary to decrease the dimension of data to reduce the number of feature attributes, eliminate irrelevant or redundant features, and improve the accuracy of the model. The dimensionality-reduction methods used in this paper are hierarchical clustering analysis [34] and LASSO analysis [35].

1. Hierarchical clustering analysis

Cluster analysis is a commonly used data dimensionality reduction technology, which can reduce a large number of observed values into several categories. The similarity between data is determined by defining a distance or similarity coefficient. There are many methods for cluster analysis, including the partitioning method, hierarchical method, density-based method, etc. The method adopted in this paper was hierarchical clustering [36]. The hierarchical clustering method was adopted without supposing the number of clusters, and Euclidean distance was used to calculate the similarity between different data points.

2. LASSO analysis

Least absolute shrinkage and selection operator (LASSO) analysis is a method that can establish a generalized linear model and screen variables. The purpose of introducing this model is to effectively screen features with multicollinearity, reduce data dimensions, and retain feature vectors that can accurately represent the features of input data. Its definition is as follows:

$$\chi'_i = \frac{\chi_i - \chi_{min}}{\chi_{max} - \chi_{min}} \quad (2)$$

$$\min_{\xi} \frac{1}{2} \left\| \gamma_i - \sum_{i=1}^n \chi'_i \xi \right\|_2^2 + \lambda \|\xi\|_1 \quad (3)$$

where χ'_i , χ_i , χ_{max} and χ_{min} denote the normalized value, the original value, the maximum and minimum value of the input variable, and γ_i denotes the output value. λ is a non-negative regular parameter controlling the complexity of the model; the larger λ is, the more severe the punishment will be for models with more features, resulting in a model with fewer features.

2.1.3. Key Sites Identification

The dimensionality of the input data remains large after the data reduction, but some inputs remain stable throughout the run cycle and do not contribute to the output value in the process model. Therefore, the Olden method (connection-weight method) [37] was used in this work to analyze the sensitivity of the input after the dimensionality reduction of data. Sites with basically unchanged data or small fluctuations were not used as input variables. The specific process is shown in Figure 2.

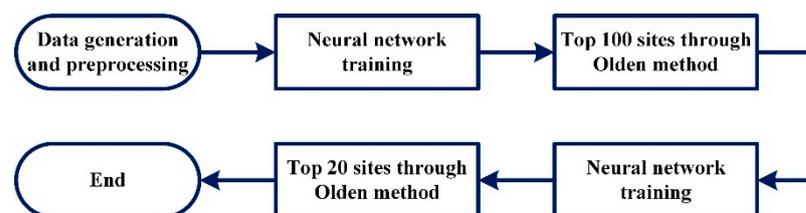


Figure 2. Sensitivity analysis procedure based on neural network.

After the original data were preprocessed and one-step prediction, 100 sites with great influence on filter pressure difference were obtained by Olden method. The 100 sites were used as inputs to rebuild the neural network, and the top 20 sites with large weights were collected as key control sites.

2.2. RNN-LSTM Model Construction

A recurrent neural network is a structure that repeats itself over time. It has been widely used in many fields, such as natural language processing (NLP), and in speech and images [29]. In the loop structure shown in Figure 3, module A of each neural network reads some input x_t and outputs a value h_t , and then repeats the loop. Loops allow information to be passed from the current step to the next.

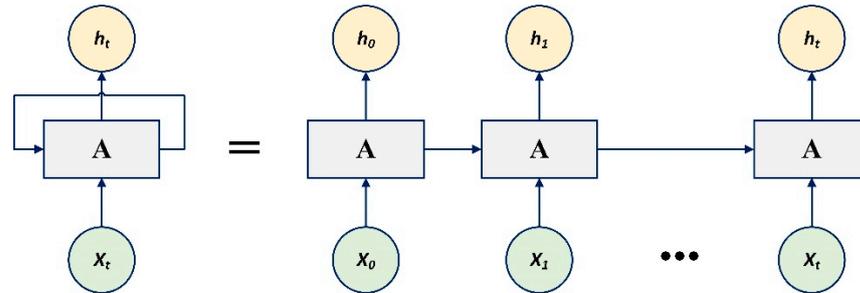


Figure 3. Graphical illustration of RNN structures.

Although RNN is suitable for time-series data-processing, it may cause gradient disappearance or gradient inflation when calculating the relationship between distant nodes in a time series. LSTM structures can learn to keep only relevant information to make predictions and forget irrelevant data. LSTM is an enhanced RNN structure. The hidden-layer information of the RNN moment only comes from the current input and the hidden layer information of the previous moment and has no memory function. The LSTM structure is shown in Figure 4.

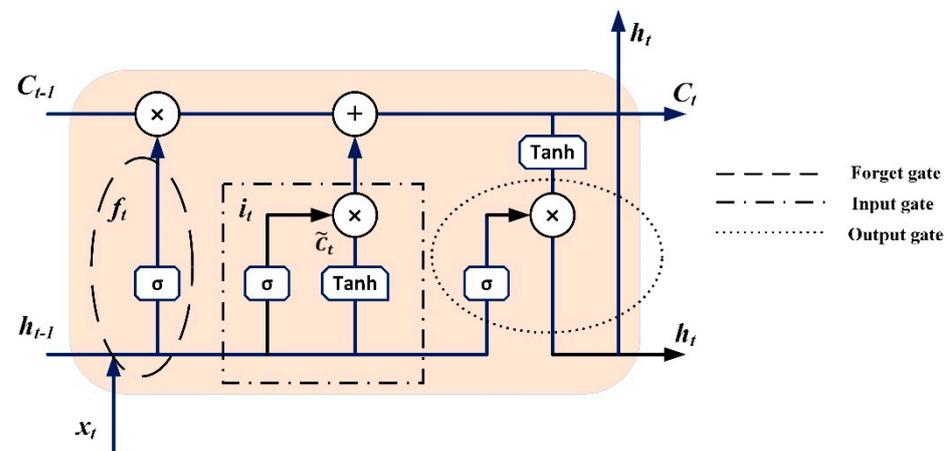


Figure 4. Graphical illustration of LSTM structures.

In the picture above, x_t denotes the input, h_{t-1} is the output of the previous layer, f_t is the output of forget gate, σ denotes sigmoid function, i_t is the output of input gate, \tilde{C}_t is the output of candidate gate, \tanh denotes the hyperbolic tangent function, C_t is the output of memory gate.

The first step in LSTM is to decide what information to forget from the cell state. This decision is made through the forget gate. The forget gate reads the previous output h_{t-1} and the current input x_t , makes a nonlinear mapping of sigmoid, and outputs a vector f_t , which is finally multiplied by the cell state C_{t-1} .

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (4)$$

where W_f is the weight matrix of the forget gate, b_f is the bias of the forget gate.

The input gate uses the sigmoid function to control which new information can be added in the current moment of the network. The sigmoid layer called the input gate layer determines what values update. The tanh layer creates a new candidate value vector \tilde{C}_t that will be added to the cell state.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (5)$$

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (6)$$

where W_i is the weight matrix of input gate, b_i is the bias of input gate, W_c is the weight matrix of forget gate, b_c is the bias of forget gate.

The cell state can be updated to

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (7)$$

Finally, the value for the output is determined. This output will be based on cell state shown in Equation (7). The cell state is processed through tanh (to get a value between -1 and 1) and multiplied by the output of the sigmoid gate to get the output.

$$o_t = \sigma(W_o [h_{t-1}, x_t] + b_o) \quad (8)$$

$$h_t = o_t * \tanh(C_t) \quad (9)$$

where o_t is the output of output gate, W_o is the weight matrix of output gate, b_o is the bias of output gate.

2.3. MPC Using RNN-LSTM Model

The core contents of model predictive control are the process variable and state predictions based on the model rolling optimization strategy and online error correction based on feedback information. The simple flow of model predictive control is shown in Figure 5.

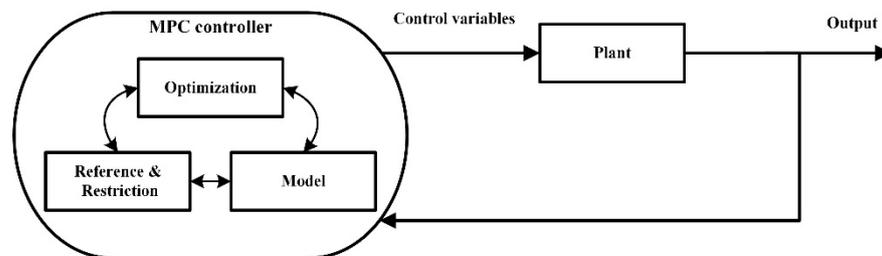


Figure 5. Schematic diagram of MPC.

2.3.1. Model Prediction

Since the corn-to-sugar process is a multi-input and single-output (MISO) system, the input vector is expressed as $X = [x_1 x_2 \dots x_n]^T$. The following equations are all based on discrete time. At time k , if the control action (input vector) is constant, the predicted value of the output can be obtained by the neural network model.

$$\hat{y}_{0,(k+j)} = f\left(X_{(k+j)}, y_{0,(k+j-1)}\right) = f\left(x_{1,(k+j)}, x_{2,(k+j)}, x_{n,(k+j)}, y_{0,(k+j-1)}\right) \quad (10)$$

$$j = 1, 2, \dots, P$$

where $f(\cdot)$ is an input/output mapping relationship represented by the RNN model, P is the prediction step, also the optimization time domain, y_0 is the actual output values of the factory (in this case, dextrose equivalent value, which will be discussed in Section 3), and \hat{y}_0 is the predictive output values of the plant model.

Assuming from time k there are M continuous control increments $\Delta u_{(k+1)}, \Delta u_{(k+2)}, \dots, \Delta u_{(k+M)}$, where each control increment can be expressed as:

$$\Delta u_{(k+M)} = [\Delta u_1 \Delta u_2 \dots \Delta u_n]^T_{(k+M)} \tag{11}$$

where, M is the control time domain. Generally, the first few steps of the control behavior determine the majority of the control results, and the remaining control steps have smaller influence. In order to save computational resources, the relationship between M value and P value is usually set as $0.1P \leq M \leq 0.2P$ [38,39]. P and M are set to be equal in this paper to consider the influence of all control steps.

When control incremental are applied to the system, the output prediction sequence value is $\hat{y}_{1,(k+1)}, \hat{y}_{1,(k+2)}, \dots, \hat{y}_{1,(k+P)}$. Note that $\hat{y}_{0,(k+j)}$ denotes the predicted output value of the RNN model without control action, while $\hat{y}_{1,(k+j)}$ denotes the predicted output value of the control system. The predicted value $\hat{y}_{1,(k+j)}$ is correlated with $\Delta u_{(k+M)}$;

In order to make the calculation more convenient, we assume that the system can be regarded as a linear system at the working point, then the system will satisfy both superposition property and uniformity. The superposition property means that when several input signals act on the system together, the total output is equal to the sum of the output generated by each input acting separately. Uniformity means that when the input signal increases by several times, the output correspondingly increases by the same multiple. The linearization method is to expand the nonlinear function into Taylor series at the equilibrium point, then take the linear terms (0 order terms and 1 order terms), ignore the higher order terms, and then get a linear mathematical model. Therefore, assuming that the control input output caused by changes can be linear superposition, Equation (10) can be represented to

$$\hat{y}_{1,(k+j)} = \hat{y}_{0,(k+j)} + a_j \Delta u_{(k+j)} \tag{12}$$

where, a_j is the step response coefficient vector, $a_j = [a_{j,1} a_{j,2} \dots a_{j,n}]$.

The vector a_j is obtained by the following local linearization method. At time k , keep all other input variables and model parameters unchanged, and only input variables $u_r (r = 1, 2, \dots, n)$ add a unit incremental value, then the neural network model is used to predict and calculate P output values $\hat{y}_{r,(k+1)}, \hat{y}_{r,(k+2)}, \dots, \hat{y}_{r,(k+P)}$. At time k , the step response sequence of control variable u_r be $\{\hat{y}_{r,(k+1)} - \hat{y}_{0,(k+1)}, \hat{y}_{r,(k+2)} - \hat{y}_{0,(k+2)}, \dots, \hat{y}_{r,(k+P)} - \hat{y}_{0,(k+P)}\}$. Element $a_{i,r} = (\hat{y}_{r,(k+i)} - \hat{y}_{0,(k+i)})$ is the i th response coefficient of the control variable u_r . It forms the step response matrix at time k

$$A(k) = \begin{bmatrix} a_1 & 0 & \dots & 0 \\ a_2 & a_1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_P & a_{P-1} & \dots & a_{P-M+1} \end{bmatrix} \tag{13}$$

$$= \begin{bmatrix} a_{11}a_{12} \dots a_{1n} & 0 & \dots & 0 \\ a_{21}a_{22} \dots a_{2n} & a_{11}a_{12} \dots a_{1n} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{P1}a_{P2} \dots a_{Pn} & a_{(P-1),1}a_{(P-2),2}a_{(P-1),n} & \dots & a_{(P-M+1),1}a_{(P-M+1),2} \dots a_{(P-M+n),n} \end{bmatrix} \tag{14}$$

The predicted output vector of the uncontrolled increment is

$$\hat{Y}_{0,(k)} = [\hat{y}_{0,(k+1)} \quad \hat{y}_{0,(k+2)} \quad \dots \quad \hat{y}_{0,(k+P)}]^T \tag{15}$$

The control increment vector is

$$\Delta U_{M,(k)} = \left[\Delta \mathbf{u}_{(k+1)} \quad \Delta \mathbf{u}_{(k+2)} \quad \cdots \quad \Delta \mathbf{u}_{(k+M)} \right]^T \tag{16}$$

The output prediction vector with control increment is

$$\hat{Y}_{1,(k)} = \left[\hat{y}_{1,(k+1)} \quad \hat{y}_{1,(k+2)} \quad \cdots \quad \hat{y}_{1,(k+P)} \right]^T \tag{17}$$

Therefore, the output prediction vector is

$$\hat{Y}_{1,(k)} = \hat{Y}_{0,(k)} + A_{(k)} \cdot \Delta U_{M,(k)} \tag{18}$$

2.3.2. Rolling Optimization

Assuming the desired output value of P future times be $w_{(k+1)}, w_{(k+2)}, \dots, w_{(k+P)}$. The purpose of optimization is to make the predicted output of the future P moments $\hat{y}_{1,(k+i)}$ as close to the expected value $W_{(k+i)}$ as possible. Define the performance indicator as Equation (19).

$$\min J_{(k)} = \sum_{i=1}^P \|q_i \cdot (w(k+i) - \hat{y}_M(k+i))\|^2 + \sum_{j=1}^M \|r_j \cdot \Delta \mathbf{u}(k+j)\|^2 \tag{19}$$

where, q_i is the tracking error weight coefficient, and r_j is the control quantity suppression weight coefficient.

In order to avoid the drastic change of the output of the controlled system, the expected reference trajectory of the output value of the system is adopted in the form of exponential change.

$$w(k+i) = \alpha^i y(k) + (1 - \alpha^i) y_r \tag{20}$$

where, α is the parameter of the reference trajectory; $y(k)$ is the actual sampling value at the current time; y_r is the target value set. The smaller the value of α , the shorter the time it takes for the reference trajectory to reach the target value.

From $\frac{\partial J_{(k)}}{\partial \Delta U_{M,(k)}} = 0$, substituting Equation (18) into Equation (19), we can figure out

$$\Delta U_{M,(k)} = \left(A^T Q A + R \right)^{-1} A^T Q \left[W_{P,(k)} - \hat{Y}_{0,(k)} \right] \tag{21}$$

where Q is the tracking error weight coefficient matrix, $Q = \text{diag}(q_1, q_2, \dots, q_P)$. R is the inhibition weight coefficient matrix of control quantity, $R = \text{diag}(r_1, r_2, \dots, r_M)$.

If boundary constraints are not considered and the first control increment $\Delta \mathbf{u}_{(k+1)}$ is taken as the actual imposed control, then $X_{k+1} = X_k + \Delta \mathbf{u}_{(k+1)}$ can be obtained. Continue solving to get $\Delta U_{M,(k+1)}$, and apply control $X_{k+2} = X_{k+1} + \Delta \mathbf{u}_{(k+2)}$. Finally, Equation (22) can be obtained

$$\Delta \mathbf{u}_{(k+1)} = C^T \cdot \Delta U_{M,(k)} \tag{22}$$

where $C = [100 \dots 0]^T$.

2.3.3. Error Correction

The predicted output sequence of the facility with incremental control is $\hat{y}_{1,(k+1)}, \hat{y}_{1,(k+2)}, \dots, \hat{y}_{1,(k+N)}$. At the next time of $k+1$, one obtains the sampling output $y_{1,(k+1)}$. Define the prediction error at time k

$$e_{(k)} = y_{(k+1)} - \hat{y}_{(k+1)} \tag{23}$$

The N prediction errors before time k are denoted in vector form

$$\mathbf{e}_{N,(k)} = \left[e_{(k-N)} e_{(k-N+1)} \cdots e_k \right]^T \tag{24}$$

The feedback error can be used to correct the prediction error as well as the prediction model. Taking the corrected prediction error as an example, the uncorrected prediction vector is defined as

$$\hat{Y}_{1,N,(k)} = \left[\hat{y}_{1,(k+1)} \quad \hat{y}_{1,(k+2)} \quad \cdots \quad \hat{y}_{1,(k+N)} \right]^T \quad (25)$$

The corrected prediction vector is

$$\hat{Y}_{cor,(k)} = \hat{Y}_{1,N,(k)} + H \cdot e_{N,(k)} \quad (26)$$

where $H = [h_1 \ h_2 \dots h_N]$, which is called error correction weight coefficient vector.

After the horizon is rolled forward by one time period, the error of the sampled output is first calculated by Equation (23). Based on the calculation of error, the prediction of the correction vector is updated according to Equation (26). In the next step of optimization calculation according to Equation (21), $\hat{Y}_{0,(K)}$ for calculating the optimal control input adopts the corrected prediction vector $\hat{Y}_{cor,(K)}$ [40,41].

3. Corn-to-Sugar Process Application

In this paper, model predictive control method based on a recurrent neural network and LSTM model was applied to corn-to-sugar factory. The flow chart of this plant is shown in Figure 6. The whole process includes a starch section, fructose section, citric acid section and starch-saccharification section. Modeling and predictive control were carried out for the starch-saccharification section, which is the core of the corn-to-sugar plant. Corn is grounded to produce a series of high-value products such as starch and protein powder. Corn starch can then be hydrolyzed by further amylase to glucose, which is converted to fructose by isomerase. Based on the existing process flow of corn-to-sugar factory, we collected the raw materials and product analysis and testing data accumulated in the factory for a long time were recorded by a DCS system, then established the production process simulation and prediction system, and carried out real-time optimization control of the production process.

The dextrose equivalent (DE) value refers to the percentage of reducing sugar in syrup in its dry matter, and it is the most important parameter to evaluate the saccharification effect. As it is an important indicator of the final product quality, the DE value was used as the control system output. Due to the different degrees of hydrolysis of starch syrup, the composition and properties of various products have different DE values. The DE value can be calculated through Equation (27). DE is a defining equation calculated directly from the factory data.

$$DE = \frac{RSC}{DMC \cdot SRD} \quad (27)$$

where RSC denotes the reducing sugar content, DMC denotes the dry matter content, SRD denotes the relative density of the sugar solution. These can be collected from corn-to-sugar factory.

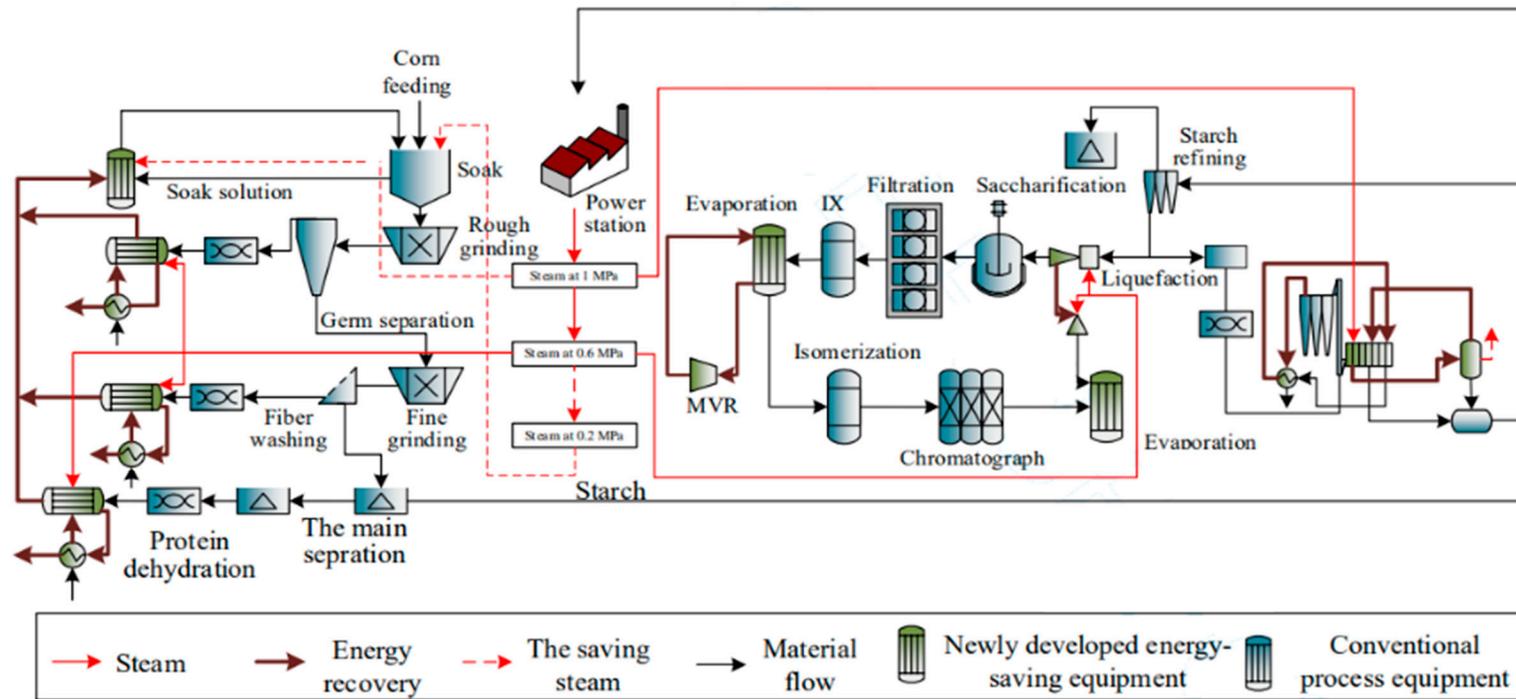


Figure 6. Flow chart of corn-to-sugar process.

3.1. Model Training

The model established in this paper was the RNN-LSTM model, which is more practical for discrete time systems that need to adapt to different operating conditions. This section consists of two parts: first, the original data were collected and preprocessed, then the model was trained, and the accuracy of the model prediction was evaluated.

3.1.1. Data Collection and Preprocessing

We collected raw materials, product analysis and test data and equipment, running state test data recorded by the DCS system, which were accumulated at 655 sites, including maize feeding, soaking, liquefaction, saccharification, and other processes used in a corn-to-sugar factory in China. A total of 13,608 sampling points were collected, and therefore the data dimension of the database was 655*13,608.

First, noise reduction and outlier processing were carried out on the collected data, and then LASSO dimension reduction analysis was carried out. The data dimension became 114*5735. On the basis of LASSO analysis, a sensitivity analysis based on a neural network was carried out to further reduce the dimension of data to 20*5735 [42]. However, some of these sites cannot be easily manipulated in the factory. In order to better conduct the training of the RNN-LSTM model and continue the predictive control of the model, we took the 18 sites with the greatest influence on the DE value as the input variable, and the DE value as the output variable. Information about the 18 sites is shown in Table 1.

Table 1. The name and meaning of the top 18 selected sites.

Name	Meaning
CURRENT1\CIA_15207	Current of 7# fine grinding facility (A)
FIC2104_1	Outlet flow rates of clean saccharification fluid (kg/h)
PID1\LIC1401_2_5-PV	Liquid level of 5# soaking tank (mm)
15.05 (Be)	Degerming feed concentration in the first grand
Flow rates of glucoamylase (g/min)	Flow rates of glucoamylase(g/min)
Acidity of old acid (%)	Acidity of old acid (%)
PRESSURE1\PIA_2110_4	Pressure of 4# starch induced draft fan (kPa)
The dry matter	The dry matter content of liquefied liquid (%)
CURRENT1\CIA_1569_5	Current of 5# fiber dehydration rotating sieve
PRESSURE1\PIA_2112_3	Pressure of 3# starch scraper conveying wind (kPa)
PID1\LIC_302A-PV	Liquid level of three-effect evaporator condensate water in set 1 (mm)
PRESSURE1\PIA_2001_1	Pressure of 1# fiber dryer (kPa)
V1102 (5.5-6.2)	pH value of starch emulsion
Fragment of grain (%)	Fragment of grain (%)
TEMPR1\TE_2001_6	Temperature of exhaust gas in drying section (K)
FIC2104_2	Outlet flow rates of turbid saccharification fluid (kg/h)
PID1\FIC_2001_3-PV	The flowrates of 3# fiber dryer (g/s)
FLOW2\FIA_1639	The flowrates of Level 12 washing step (g/s)

3.1.2. Training Process

The data were divided into a training set and a test set, with a ratio of 9:1. The RNN model structure consists of an input layer, an LSTM layer, a dropout layer, and a fully connected layer. The activation function of LSTM layer and fully connected layer adopts tanh. The LSTM layer has 200 hidden units and selects information to be remembered in the long term or forgotten by three logic gates. The dropout probability of the dropout layer is 0.1, which can greatly avoid the occurrence of overfitting. In the training process, the mini-batch size was set to 128, the Adam optimizer was used, and the initial learning rate was 0.005; the training was stopped after 130 epochs.

Since the prediction of DE value is a regression task, we chose mean square error (MSE) as the loss function, and the calculation method is shown in Equation (28).

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (P_i - Y_i)^2 \quad (28)$$

where P_i and Y_i are the predicted value and the experimental value of the i th data and n is the total number of samples. In the trained model, MSE value was 0.02.

The training process of the model is shown in Figure 7. The curve in the figure is the loss decline curve. It can be seen from the figure that the loss function becomes stable at the 130th epochs.

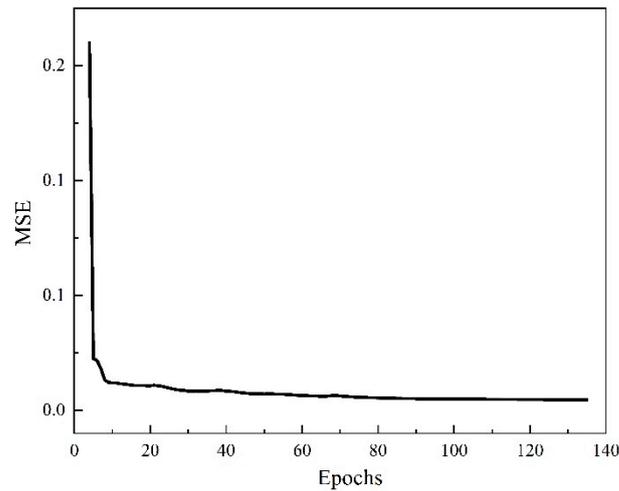
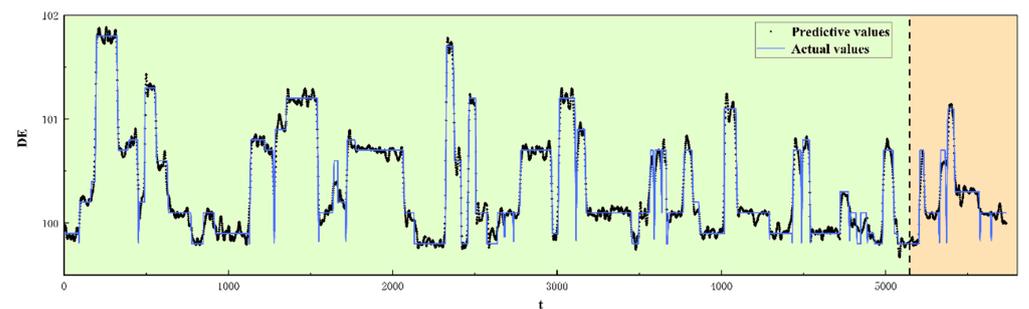
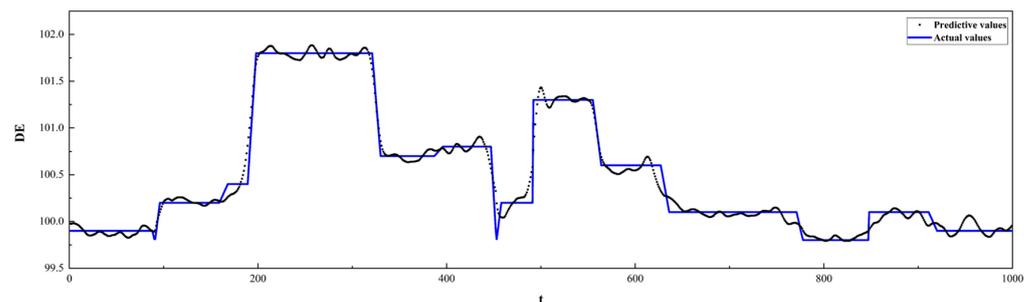


Figure 7. Loss curve of RNN-LSTM.

Figure 8a shows the DE values actually collected by the factory and the predicted values and Figure 8b is the locally enlarged view of the first 1000 sample points in Figure 8a. The first 90% of the sample points are the training set, and the last 10% are the test set. The solid line is the actual value, and the dashed line is the predicted value. It can be seen that the RNN-LSTM model has a good training effect and satisfactory prediction of the process.



(a) 5000 sample point



(b) 1000 sample point

Figure 8. Comparison of actual and predicted values of DE values.

3.2. MPC with RNN-LSTM Model

After the factory's RNN-LSTM model was built, and the output of the forecast model was corrected by feedback, a complete predictive control system was built based on the model predictive control theory [41]. The range of the manipulated variables of the predictive control system is from $0.9 \times \text{minimum value}$ to $1.1 \times \text{maximum value}$. We conducted regulations at various time points to verify the effectiveness of the control model. We used two cases to illustrate the controller performance. For case (a), the initial regulation time t_0 is at sampling point of 4000, and it is at sampling point of 4250 for case (b). In all cases, DE value is the controlled variable. In case 1 and case 2, all 18 variables in Table 1 are manipulated variables. No disturbance variables explicitly considered in these cases. Similarly, the manipulated variables in case 3 and case 4 are also the 18 variables in Table 1, but random noise is added to the output of the RNN-LSTM model in case 3 and case 4. When the RNN-LSTM model was used as a controller model to predict and calculate the optimal control input, we added noise disturbance to the output of the model prediction to simulate the disturbance of the environment to the system. The disturbance variable in case 5 is the dry-matter content of liquefied liquid, and the manipulated variables are the 17 variables in Table 1, except the disturbance variable. The disturbance variables in case 6 are the dry-matter content of liquefied liquid and fragments of grain. The manipulated variables are the 16 variables in Table 1 except the interference variables. The details about the variables can be seen in Table 2.

Table 2. List of the manipulated variables, controlled variables and disturbance variables.

	Manipulated Variables	Controlled Variables	Disturbance Variables
Case 1	all 18 variables	DE value	None
Case 2	all 18 variables	DE value	None
Case 3	all 18 variables	DE value	Random Disturbance
Case 4	all 18 variables	DE value	Random Disturbance
Case 5	17 variables	DE value	Dry matter content of liquefied liquid
Case 6	16 variables	DE value	Dry matter content of liquefied liquid and fragment of grain

If the system is not controlled and the input variables of the system remain unchanged, the LSTM neural network model is used to predict that the output of the system will increase and then become stable from time 4000, as shown in in the dashed line of Figure 9 which is a fragment of Figure 8. From time 4250, the system output will go down a little, bounce back, and then become stable. For case (b) it could be viewed as a setpoint change from 100 to 101 as the target value.

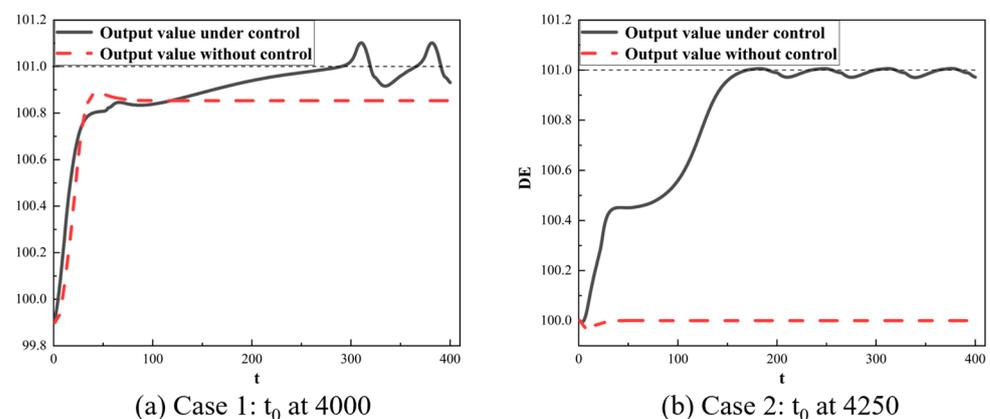


Figure 9. The state profiles for different time of case 1 and case 2.

After controlling the system, the system output will reach the target value ($DE = 101$) after 290 timesteps in case 1 and 170 timesteps in case 2, as shown in Figure 9.

In case 1, the predictive control system parameters are set as: $Q/R = 100$ ($Q = 0.6$, $R = 0.06$). In case 1, the trend of the output value which remained steady at 100.8 after 100 timesteps was changed, and the output value reached the target value within 200 timesteps by predictive control. In case 2, the predictive control system parameters were set as: $Q/R = 63$ ($Q = 7$, $R = 0.11$), the downward trend of system output was changed, and the system output value reached the target value within 170 timesteps. When the system is stable, the maximum deviation between the system output value and the target value in the two cases is 0.100 and 0.006, the control precision and the deviation are satisfactory. Generally speaking, the higher of Q/R ratio, the faster a control response becomes, and the higher risk it is for the system to be unstable.

In order to verify the applicability and practicability of the LSTM, we compared the control effect of the LSTM with CNN and ANN as the control model for case1. The comparison of simulation results is shown in Figure 10. For ANN as the control model, within the control of 400 timesteps, the system cannot rapidly generate response under the control of MPC and cannot reach the target value, indicating that the response speed of ANN model is sluggish, and the response time is unacceptable. For CNN as the control model, the response time is 289, and it is almost the same with the response time of LSTM model which is 285. However, the system overshoot $\sigma\%$ is 0.47%, which is 4.7 times as much as the $\sigma\%$ of LSTM model. Excessive overshoot may cause the system to oscillate, making the system lose stability. In the corn-to-sugar plant, excessive overshoot may cause damage to machinery. Therefore, compared with ANN and CNN, LSTM has a faster response speed and better stability.

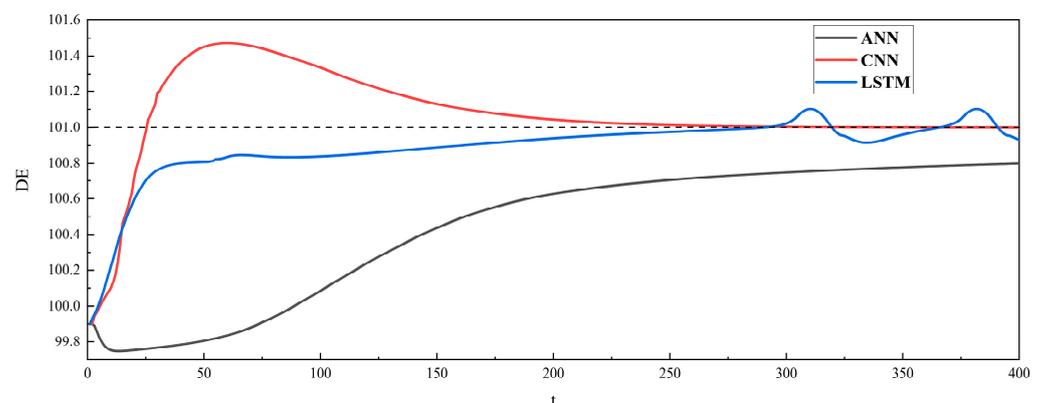


Figure 10. The state profiles of different neural network models.

In reality, the prediction model of the MPC could not fully predict the actual situation in the simulation process. However, we used the same model for control and simulation in both case 1 and case 2. In order to simulate this discrepancy, we used different RNN-LSTM models for the control model and simulation model. For the simulation model, we changed the hyperparameters of the RNN-LSTM model by including fewer units in the hidden layer and retrained a new RNN-LSTM model. For the control model, we used the data in the previous 2500 periods just before the initial regulation time t_0 , instead of the entire dataset. To simulate the accidental error in the real situation, we added a normal error representing the noise with a mean value of 0 and standard error of 0.01. We still simulated the regulation at time 4000 and 4250, respectively. The relevant situation is shown in Figure 11. After controlling of the system, the system output will reach the target value after a smaller number of steps, compared to Figure 9. This is because we used 100 hidden units in the hidden layer of RNN-LSTM for the simulation model, which are fewer than the control model, leading to a process that responds more quickly.

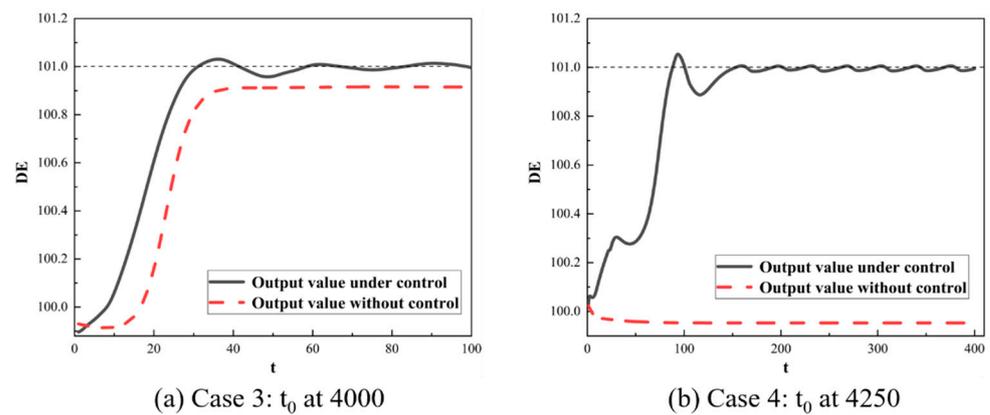


Figure 11. The state profiles for different time of case 3 and case 4.

For some practical problems with disturbances variables, we further simulated some experiments on the basis of case 3. We made the 8th variable (dry-matter content of liquefied liquid) as disturbance variable, and the rest of the 17 variables in Table 1 as the manipulated variables. When the system output value reached stability, we gave a step change of the disturbance variable 10% and kept it at this level. The simulation results are shown in Figure 12a. It can be seen that the controlled variable of the system will recover after a brief decline and continue to stabilize around the target value. From the simulated experiment, we can find the MPC strategy can resistance disturbance to a certain extent. Then, we revisited case 3 for a more complex situation. We still used the 8th variable to simulate the disturbance, and kept the 14th manipulated variable (fragment of grain) value fluctuating by 5% near the initial value to simulate the situation that the input control could not act on the system well enough and simulated it 4000 times. After the output value was stable, the target value 101 was changed to 100.6 to simulate the change of the setpoint. The simulation results are shown in Figure 12b. Due to the uncontrollable input, the system output always has small fluctuations, but the trend was that the system output reached near the target value and remains relatively stable.

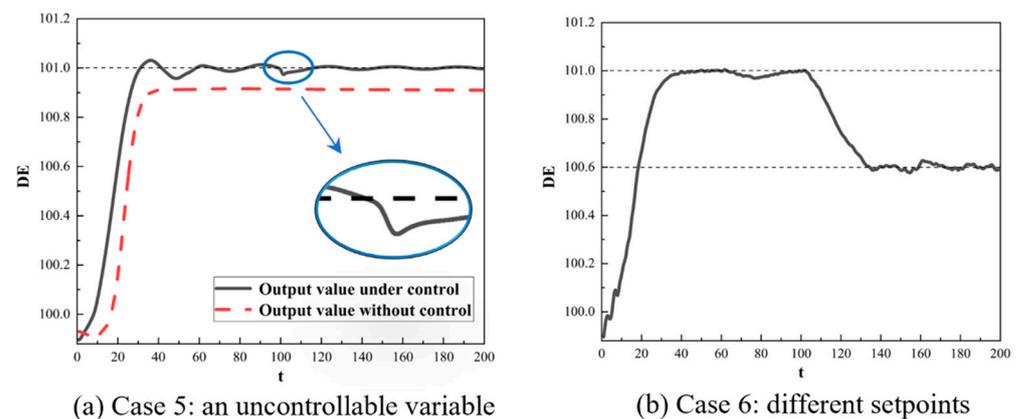


Figure 12. The state profiles for uncontrollable variables and different setpoints.

We must also comment on the tuning of the control parameters in the objective function shown in Equation (19). In the process of the simulation, adjusting the values of tracking error weight coefficient matrix (Q) and inhibition weight coefficient matrix of control quantity (R) can change the rapidity and stability of the control system. Reducing the value of Q can make the output curve of the system become smoother to a certain extent. Increasing the value of R will make the calculated $\Delta U_{M_i(k)}$ smaller and it will take longer to adjust the output to reach the target value. In some cases, the optimal input may be too small for the MPC to control the output value near the target value, resulting in

control failure. Therefore, the value of R should not be too large in order for the output value to reach the target value more quickly. However, due to the small sample size of the training model, if the value of R is too small and the calculated $\Delta U_{M,(k)}$ is too large, there will be many manipulated variables that exceed the constraint range of the variable and are forced to give the maximum or minimum value corresponding to the variable, and then the stability of the system will decrease.

The parameters of the MPC will not only affect the performance of MPC control, but also affect the computational complexity of the MPC. The parameters of the MPC used in this work mainly included the optimization time domain P , control time domain M , tracking error weight coefficient Q , and control quantity suppression weight coefficient R . If P is too large, the system will react slowly, and the controller will be difficult to control the system in time. However, if P is too small, the system will carry out a large number of optimization calculations, resulting in a large consumption of computing resources. Through trial-and-error method, the P of MPC in this work is 50. The selection of control time domain M has been described in Section 2.3.1. The M of MPC in this work is set as 50. Q and R is often set by following rules: If $Q > R$, it means that the MPC optimization strategy is more likely to reduce the difference between the target value and the controlled output value. If $R > Q$, it means that the optimization strategy of MPC would prefer that the manipulated variables vary smaller and smoother. In addition, the effects of these two parameters on the controlled output have been described above, the set value of parameter can be adjusted in detail according to the rules mentioned above and the trial-and-error method based on the demand for controlled output. We have included the Matlab code for MPC controller in our supporting information.

4. Conclusions

In this work, the application of an MPC based on an RNN-LSTM model in the process of corn-to-sugar was studied. First, the corn-to-sugar process was modeled, and the control effects of different models were compared. The results showed that the LSTM model was more suitable for modeling the corn saccharification plant, and the MSE was 0.02, which means that the model has a good prediction effect. Based on the RNN-LSTM model, the predictive model of the control system was established, the reasonable objective function was selected, and the control system was constructed by controlling the key points of the corn-to-sugar factory and combining them with feedback correction. The control system was simulated by Matlab. With proper tuning of the MPC objective parameters, the output value of the dextrose equivalent was achieved at the target, and the system was controlled well after setpoint changes or disturbances were introduced.

This work presents an overall design of MPC system based on the data-driven RNN-LSTM model, and the theoretical control effect of model predictive control in this corn-sugar production process is verified in the simulations. This work builds a solid support for a neural network-based model predictive controller at the factory-process level, and we will further verify its real-life application in future work.

Author Contributions: Introduction: J.T., Y.L. and Y.T.; Framework: Y.W., L.Z. and Y.D.; Application: J.D., J.M. and C.L.; Software: J.M. and C.L.; Data curation: J.T., Y.L., Y.T. and Y.W.; Writing—original draft preparation: J.M. and C.L.; Writing—review and editing: Y.D., J.D. and L.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Special Foundation for State Major Basic Research Program of China (Grant No. 2021YFD2101005).

Data Availability Statement: Partial data supporting the results of this study may be obtained from the corresponding author upon reasonable request.

Acknowledgments: The authors are grateful for the financial supports of Special Foundation for State Major Basic Research Program of China (Grant No. 2021YFD2101005).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Rose, D.J.; Inglett, G.E.; Liu, S.X. Utilisation of corn (*Zea mays*) bran and corn fiber in the production of food components. *J. Sci. Food Agric.* **2010**, *90*, 915–924. [[CrossRef](#)] [[PubMed](#)]
2. Venkat, A.N.; Hiskens, I.A.; Rawlings, J.B.; Wright, S.J. Distributed MPC Strategies with Application to Power System Automatic Generation Control. *IEEE Trans. Control. Syst. Technol* **2008**, *16*, 1192–1206. [[CrossRef](#)]
3. Tang, X.L.; Jia, T.; Hu, X.S.; Huang, Y.J.; Deng, Z.W.; Pu, H.Y. Naturalistic Data-Driven Predictive Energy Management for Plug-In Hybrid Electric Vehicles. *IEEE Trans. Transp. Electrification* **2021**, *7*, 497–508. [[CrossRef](#)]
4. Beal, C.E.; Gerdes, J.C. Model Predictive Control for Vehicle Stabilization at the Limits of Handling. *IEEE Trans. Control. Syst. Technol.* **2013**, *21*, 1258–1269. [[CrossRef](#)]
5. Hrovat, D.; Di Cairano, S.; Tseng, H.; Kolmanovsky, I. The development of model predictive control in automotive industry: A survey. In Proceedings of the 2012 IEEE International Conference on Control Applications (CCA), Dubrovnik, Croatia, 3 May 2012. [[CrossRef](#)]
6. Mendis, P.; Wickramasinghe, C.; Narayana, M.; Bayer, C. Adaptive model predictive control with successive linearization for distillate composition control in batch distillation. In Proceedings of the 2019 Moratuwa Engineering Research Conference (MERCon), Moratuwa, Sri Lanka, 3–5 July 2019; pp. 366–369. [[CrossRef](#)]
7. Prasad, G.M.; Kedia, V.; Rao, A.S. Multi-model predictive control (MMPC) for non-linear systems with time delay: An experimental investigation. In Proceedings of the International Conference on Measurement, Instrumentation, Control and Automation (ICMICA), Kurukshetra, India, 4–26 June 2020; pp. 1–5. [[CrossRef](#)]
8. Wang, X.; Li, S.K.; Tang, T.; Yang, L.X. Event-Triggered Predictive Control for Automatic Train Regulation and Passenger Flow in Metro Rail Systems. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 1782–1795. [[CrossRef](#)]
9. Ellis, M.; Durand, H.; Christofides, P.D. A tutorial review of economic model predictive control methods. *Process Control.* **2014**, *24*, 1156–1178. [[CrossRef](#)]
10. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [[CrossRef](#)]
11. Li, X.; Zhang, L.; Wang, Z.; Dong, P. Remaining useful life prediction for lithium-ion batteries based on a hybrid model combining the long short-term memory and elman neural networks. *Energy Storage* **2019**, *21*, 510–518. [[CrossRef](#)]
12. Liang, X.; Lin, L.; Shen, X.; Feng, J.; Yan, S.; Xing, E.P. Interpretable structure-evolving LSTM. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2171–2184.
13. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)]
14. Graves, A.; Schmidhuber, J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Netw.* **2005**, *18*, 602–610. [[CrossRef](#)] [[PubMed](#)]
15. Akhter, M.N.; Mekhilef, S.; Mokhlis, H.; Ali, R.; Usama, M.; Muhammad, M.A.; Khairuddin, A.S.M. A hybrid deep learning method for an hour ahead power output forecasting of three different photovoltaic systems. *Applied Energy* **2022**, *307*. [[CrossRef](#)]
16. Demidova, L.A. Recurrent Neural Networks' Configurations in the Predictive Maintenance Problems. *Workshop Mater. Eng. Aeronaut.* **2020**, *714*, 012005. [[CrossRef](#)]
17. Xu, J.; Li, C.D.; He, X.; Huang, T.W. Recurrent neural network for solving model predictive control problem in application of four-tank benchmark. *Neurocomputing* **2016**, *190*, 172–178. [[CrossRef](#)]
18. Norouzi, A.; Shahpouri, S.; Gordon, D.; Winkler, A.; Nuss, E.; Abel, D.; Andert, J.; Shahbakhti, M.; Koch, C.R. Machine Learning Integrated with Model Predictive Control for Imitative Optimal Control of Compression Ignition Engines. *IFAC-PapersOnLine* **2022**, *55*, 19–26. [[CrossRef](#)]
19. Maltais, L.G.; Gosselin, L. Energy management of domestic hot water systems with model predictive control and demand forecast based on machine learning. *Energy Convers. Manag.* **2022**, *15*, 100254. [[CrossRef](#)]
20. 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](#)]
21. Wu, Z.; Christofides, P.D. Control Lyapunov-Barrier function-based predictive control of nonlinear processes using machine learning modeling. *Comput. Chem. Eng.* **2020**, *143*, 106706. [[CrossRef](#)]
22. Wu, Z.; Rincon, D.; Christofides, P.D. Real-time machine learning for operational safety of nonlinear processes via barrier-function based predictive control. *Chem. Eng. Res. Des.* **2020**, *155*, 88–97. [[CrossRef](#)]
23. Alhajeri, M.S.; Luo, J.W.; Wu, Z.; Albalawi, F.; Christofides, P.D. Process structure-based recurrent neural network modeling for predictive control: A comparative study. *Chem. Eng. Res. Des.* **2022**, *179*, 77–89. [[CrossRef](#)]
24. Ren, Y.M.; Alhajeri, M.S.; Luo, J.W.; Chen, S.R.; Abdullah, F.; Wu, Z.; Christofides, P.D. A tutorial review of neural network modeling approaches for model predictive control. *Comput. Chem. Eng.* **2022**, *165*, 107956. [[CrossRef](#)]
25. Zarzycki, K.; Lawrynczuk, M. LSTM and GRU Neural Networks as Models of Dynamical Processes Used in Predictive Control: A Comparison of Models Developed for Two Chemical Reactors. *Sensors* **2021**, *21*, 5625. [[CrossRef](#)] [[PubMed](#)]
26. Wang, Q.C.; Pan, L.; Lee, K.Y. Improving Superheated Steam Temperature Control Using United Long Short-Term Memory and MPC. *IFAC-PapersOnLine* **2020**, *53*, 13345–13350. [[CrossRef](#)]
27. Singh, A.K.; Tyagi, B.; Kumar, V. Classical and Neural Network-Based Approach of Model Predictive Control for Binary Continuous Distillation Column. *Chem. Prod. Process Model.* **2014**, *9*, 71–87. [[CrossRef](#)]
28. Qing, X.Y.; Song, J.; Jin, J.; Zhao, S.L. Nonlinear model predictive control for distributed parameter systems by time-space-coupled model reduction. *AIChE J.* **2021**, *67*, e17246. [[CrossRef](#)]

29. Lin, W.C.; Tsai, C.F. Missing Value Imputation: A Review and Analysis of the Literature (2006–2017). *Artif. Intell. Rev.* **2020**, *53*, 1487–1509. [[CrossRef](#)]
30. Strike, K.; El Emam, K.; Madhavji, N. Software cost estimation with incomplete data. *IEEE Trans. Softw. Eng.* **2001**, *27*, 890–908. [[CrossRef](#)]
31. Chang, Z.; Zhang, Y.; Chen, W. Electricity price prediction based on hybrid model of adam optimized LSTM neural network and wavelet transform. *Energy* **2019**, *187*, 115804. [[CrossRef](#)]
32. Maples, M.P.; Reichart, D.E.; Konz, N.C.; Berger, T.A.; Trotter, A.S.; Marti, J.R.; Dutton, D.A.; Paggen, M.L.; Joyner, R.E.; Salemi, C.P. Robust Chauvenet Outlier Rejection. *Astrophys. J. Suppl. Ser.* **2018**, *238*, 2. [[CrossRef](#)]
33. Elko, G.W.; Sondhi, M.M.; West, J.E. Noise reduction processing arrangement for microphone arrays. *Acoust. Soc. Am.* **1989**, *88*, 2919. [[CrossRef](#)]
34. Lopez-Medina, C.; Ladehesa-Pineda, L.; Puche-Larrubia, M.A.; Escudero-Contreras, A.; Font-Ugalde, P.; Collantes-Estev, E. Which factors explain the patient global assessment in patients with ankylosing spondylitis A hierarchical cluster analysis on REGISPONSER-AS. *Semin. Arthritis Rheum.* **2021**, *51*, 875–879. [[CrossRef](#)]
35. Lin, J.; Li, S. Sparse recovery with coherent tight frames via analysis Dantzig selector and analysis LASSO. *Appl. Comput. Harmon. Anal.* **2014**, *37*, 126. [[CrossRef](#)]
36. MacQueen, J. Some methods for classification and analysis of multivariate observations. In Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, CA, USA, 1 January 1967; p. 281.
37. Zhang, X.; Liu, L.; Long, G.; Jiang, J.; Liu, S. Episodic memory govern choices: An RNN-based reinforcement learning model for decision-making task. *Neural Netw.* **2021**, *134*, 1–10. [[CrossRef](#)] [[PubMed](#)]
38. Ramasamy, V.; Kannan, R.; Muralidharan, G.; Sidharthan, R.K.; Veerasamy, G.; Venkatesh, S.; Amirtharajan, R. A comprehensive review on Advanced Process Control of cement kiln process with the focus on MPC tuning strategies. *J. Process. Control.* **2023**, *121*, 85–102. [[CrossRef](#)]
39. Sun, C.; Hu, X.; Moura, S.J.; Sun, F. Velocity Predictors for Predictive Energy Management in Hybrid Electric Vehicles. *IEEE Trans. Control. Syst. Technol.* **2014**, *23*, 1197–1204. [[CrossRef](#)]
40. Maciejowski, J.M. *Predictive Control with Constraints*; Prentice-Hall, Pearson Education Limited: Harlow, UK, 2002.
41. Rawlings, J.B.; Mayne, D.Q. *Model Predictive Control: Theory and Design*; Nob Hill Publishing: Madison, WI, USA, 2009.
42. Tong, Y.; Shu, M.; Li, M.X.; Liu, Y.W.; Tao, R.; Zhou, C.C.; Zhao, Y.; Zhao, G.X.; Li, Y.; Dong, Y.C.; et al. A neural network-based production process modeling and variable importance analysis approach in corn to sugar factory. *Front. Chem. Sci. Eng.* **2022**, 1–4. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.