

Article

Robust Model Predictive Controller Using Recurrent Neural Networks for Input–Output Linear Parameter Varying Systems

Mohsen Hadian ¹, Amin Ramezani ^{1,*} and Wenjun Zhang ² 

¹ College of Electrical and Computer Engineering, Tarbiat Modares University, Tehran 14117-13116, Iran; m.hadian@modares.ac.ir

² College of Engineering, Department of Mechanical Engineering, University of Saskatchewan, Saskatoon, SK S7N 5E5, Canada; chris.zhang@usask.ca

* Correspondence: ramezani@modares.ac.ir

Abstract: This paper develops a model predictive controller (MPC) for constrained nonlinear MIMO systems subjected to bounded disturbances. A linear parameter varying (LPV) model assists MPC in dealing with nonlinear dynamics. In this study, the nonlinear process is represented by an LPV using past input–output information (LPV-IO). Two primary objectives of this study are to reduce online computational load compared with the existing literature of MPC with an LPV-IO model and to confirm the robustness of the controller in the presence of disturbance. For the first goal, a recurrent neural network (RNN) is employed to solve real-time optimization problems with lower online computation. Regarding robustness, a new control law is developed, which comprises a fixed control gain (K) and a free perturbation (C). The proposed method enjoys a shrunken conservatism owing to the finding of a larger possible terminal region and using free control moves. The strategy is examined in an alkylation of benzene process and displays outstanding performance in both setpoint tracking and disturbance rejection problems. Moreover, the superiority of RNN over three conventional optimization algorithms is underlined in terms of MSE, the average time for solving the optimization problem, and the value of the cost function.

Keywords: model predictive control; robust model predictive control; recurrent neural network; linear parameter varying; LPV; MPC; RNN



Citation: Hadian, M.; Ramezani, A.; Zhang, W. Robust Model Predictive Controller Using Recurrent Neural Networks for Input–Output Linear Parameter Varying Systems. *Electronics* **2021**, *10*, 1557. <https://doi.org/10.3390/electronics10131557>

Academic Editors: Jong-Myon Kim, Hyeung-Sik Choi and Farzin Piltan

Received: 2 June 2021

Accepted: 22 June 2021

Published: 28 June 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 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

Over the past few decades, model predictive controllers (MPCs) have attracted a great deal of attention in industry and academia [1–4]. This considerable prominence of MPC is rooted in several factors, including simple generalization to a MIMO system, applicability to non-minimum phase and unstable processes, capability to compensate delay, and imposing output/input constraints [1]. While influential theories exist in MPC for linear systems, some systems are nonlinear and operate under large operating conditions, or some external parameters can fundamentally change the process response [5]. Nonlinear MPC (NMPC) has been developed to deal with systems with nonlinear dynamics. A higher level of algorithm complexity and excessive computational load are two main drawbacks of NMPC.

A linear parameter varying (LPV) framework is established in this paper to mitigate these problems by assisting the MPC to model dynamic and static nonlinearities [6–9]. MPC founded on LPV has aroused enthusiasm among researchers to ensure practical implementation to bridge the gap between linear MPC and NMPC [10–12] (less complicated than nonlinear models and more accurate than linear models). With the help of LPV, MPC can be employed for nonlinear and/or time-varying systems [6,13–15]. LPV offers MPC a linear model instead of nonlinear models, where the scheduling variable denotes nonlinear dynamic parameters, changing environmental conditions, and operating points. It is also noteworthy that, in an LPV model, current values of scheduling variables can be measured

or estimated, but the future values are unknown, so a robust MPC is required, or the future values can be estimated.

Most MPC-LPV literature is centered around the state-space model, while states might not be measurable in many industrial applications. Moreover, using an observer can complicate the design. In Reference [16], MPC-LPV with SS and IO presentations are compared for a control moment gyroscope (CMG). The results showed that the execution time would be reduced when using the IO framework. To this end, MPC is established based on an LPV-IO framework in this paper.

Only a few MPCs have been documented for LPV systems under the I/O formulation, mainly focused on stability, computation, and conservatism [5]. One of the initial research works on MPC with the LPV-IO model can be found at [10], where min-max (worse-case) cost functions are regarded. The cost function includes a stage cost for control performance and a terminal cost to grant stability. The optimization problem of MPC is written in bilinear matrix inequality (BMI), which can be non-convex and challenging. This approach was improved in [17] in order to reduce the complexity level markedly. The authors designed a controller based on a linear matrix inequality (LMI) convex problem, successfully assessed in an ideal continuous stirred tank reactor (CSTR). They also enlarge the terminal region to moderately reduce conservatism. This work employed offline controllers and terminal regions to reduce the online computational burden moderately. However, the controller still suffers from substantial online computational capacity, stemming from solving LMI equations and calculating the Markov coefficient of the LPV system [18]. Above all, it is assumed that the future values of scheduling variables can vary inside a convex polytope. This assumption is less conservative than a frozen or fixed scheduling variable over the prediction horizon while making the optimization problem nonlinear or unreachable [5,17]. The online computational complexity of [17] has been slightly ameliorated in a sequence of quadratic programs. The main drawback of this work is that it lacks a provision for recursive feasibility.

The first novelty of our paper is to make the controller, developed in [10], more computationally efficient. To the best of our knowledge, for the first time, a computationally efficient MPC with an LPV-IO model was presented in this study while keeping the conservatism degree as low as reasonably achievable. A recurrent neural network (RNN) is proposed to deal with the online optimization problem in this paper. Thanks to the global convergence and low computational burden, neural network-based optimizations have been applied to linear, quadratic [19,20], nonlinear programming [21,22], and variational inequality problems [23,24]. The RNN is pointed out to display convincing performance in either MPC or NMPC [25–27], in which nonlinear optimization can be converted to quadratic programming. The motive behind selecting RNN is its good adaptability as well as lower complexity, even in real-time nonlinear and large-scale optimization. That is why the studied method was assessed in the alkylation of benzene process, a large-scale nonlinear process with 25 state variables, 5 inputs, and 5 outputs.

The second novelty of this paper is to guarantee the robustness of the system when facing bounded disturbance. The problem of disturbance rejection has not been studied so far for MPC with LPV-IO models. A stabilizing control law based on the input–output measurements is proved to ensure the superior closed-loop performance of the controller in the presence of additive disturbance. The control law includes a fixed control gain (K) and a free perturbation (C), the former of which is designed offline with the largest possible terminal region to reduce conservatism. The free control moves are determined online to ensure input-state stability in the presence of bounded disturbances. Adding control moves provides a less conservative framework than those studied in [17,18], where only offline control gain is employed.

To sum up, the contributions of this paper are as follows:

- An RNN-based optimization algorithm is developed to offer global convergence and lower the online computational load.

- Free control moves are added to the constant control gain to maintain the closed-loop stability when facing bounded disturbances.
- Concerning previous studies for MPC with LPV, the proposed method inherently enjoys a shrunk conservatism degree as a result of finding the larger possible terminal region, using free control moves, and the global solution of the optimization problem.

The rest of the paper is organized as follows. The standard nonlinear model is reformulated as an LPV-IO model in Section 2. The proposed RMPC-LPV structure is developed in Section 3 and proved to be stable and feasible. The MPC-RNN is presented in Section 4. Next, the state-space equations, inputs, outputs, and critical parameters of the alkylation of benzene process are wholly outlined in Section 5, followed by a collection of simulations to evaluate the proposed control methodology. Key findings are finally summarized in Section 6.

2. Problem Statement

Assume the following discrete-time MIMO linear parameter-varying (LPV) transfer function with constraints:

$$y(k) = -\left(\sum_{i=1}^{n_a} A_i(p(k))q^{-i}\right)y(k) - \left(\sum_{j=1}^{n_b} B_j(p(k))q^{-j}\right)u(k) + d(p(k)) \quad (1)$$

Subject to the following constraints

$$\begin{aligned} u(k) &\in U \equiv \{u \in \mathbb{R}^{n_u} \mid |u(k)| \leq u_{max}\} \\ \Delta u(k) &\in V \equiv \{\Delta u \in \mathbb{R}^{n_u} \mid |\Delta u(k)| \leq \Delta u_{max}\} \\ y(k) &\in Y \equiv \{y \in \mathbb{R}^{n_y} \mid |y(k)| \leq y_{max}\} \end{aligned} \quad (2)$$

where $y(k)$ are outputs; $u(k)$ are inputs; $p(k)$ are scheduling variables; q^{-i} is backshift operator; n_a is the degree of the output polynomial; n_b is the degree of the input polynomial; n_u is the number of input; n_y is the number of outputs; and u_{max} , Δu_{max} , and y_{max} are boundaries. The state-space representation of the dynamic model given by Equation (1) is as follows [10]:

$$x(k+1) = \underbrace{\begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}}_{A(p(k))} x(k) + \underbrace{\begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}}_{B(p(k))} u(k) + d(k) \quad (3)$$

where $x(k) = [y(k-1) \dots y(k-n_a) \ u(k-1) \dots u(k-n_b)]^T$. For the sake of simplicity, $p(k)$ is removed from all coefficients a_i and b_i . The following assumptions are made:

A1: The parameter-varying matrix is $[A(k) \ B(k)] \in \Omega = \text{Co}[[A_1 \ B_1], \dots, [A_m \ B_m]]$, where Ω is the polytope, Co is the convex hull, and $[A_i \ B_i]$ are vertices corresponding to scheduling variable p_i :

$$p = \text{Co}\{p_1, \dots, p_m\} \quad (4)$$

A2: The profile of changes in scheduling variables is predetermined from a safety and economic point of view with the following upper and lower bound:

$$p \in P \equiv \{p \in \mathbb{R}^{n_p} \mid |p(k)| \leq p_{max}\} \quad (5)$$

A3: The disturbance is bounded:

$$d(k) \in D \equiv \{d \in \mathbb{R}^{n_d} \mid |d(k)| \leq d_{max}\} \quad (6)$$

A4: System 3 is controllable, i.e., the controllability matrix (M) has a full rank:

$$M = [B \ AB \ A^2B \ \dots \ A^{n-1}B] \quad (7)$$

3. Robust Model Predictive Controller

This paper aims to define a control law that ensures the input-to-state practical stability (ISpS) of the system (1) with constraints (2). The control law is calculated by minimizing the cost function at any time instance k :

$$\begin{aligned} V_N(x_k, u_k, p_k) = & \sum_{k=0}^{N-1} [e(k+i)^T Q e(k+i) + \Delta u(k+i-1)^T R \Delta u(k+i-1)] \\ & + [x(k+N)^T Q_p x(k+N)] \\ & \text{Subject to} \\ y(k+j) = & -\left(\sum_{i=1}^{n_a} A_i q^{-i}\right) y(k+j) - \left(\sum_{j=1}^{n_b} B_j q^{-j}\right) u(k+j) + d(k), \quad j = 1, 2, \dots, N-1 \quad (8) \\ & u(k+j) \in U \\ & \Delta u(k+j) \in V \\ & y(k+j) \in Y \\ & x(k+N) \in X_f \end{aligned}$$

where $e = r - y$, the deviation of output from the reference trajectory, and Δu is the control increment. Q and R are positive definite state and input weighting matrices in charge of closed-loop achievements, $Q_p = Q_p^T > 0$ is the terminal penalty matrix designed to ensure stability, and X_f is the terminal region. The first and second part of V_N are named stage cost (V_s) and terminal cost (V_T), respectively. Disturbances are supposed to be fixed over the prediction horizon.

According to [28], a state feedback controller can represent the system's stability with the LPV-IO form. The control law ($u(k)$) is comprised of a fixed state feedback K and a free control move (c), in which K is computed offline and c is obtained from the minimization of (8):

$$u(k) = -Kx(k) + c(k) \quad (9)$$

The state-feedback gain accounts for maintaining the final state variable $x(k+N)$ in the terminal region while meeting constraints and keeping the controller as less conservative as possible. In fact, a constant state feedback controller confirms the stability when there is no disturbance. Otherwise, free control moves (c) will be determined online in a min-max problem to guarantee the ISpS. Thus, a part of the controller will be calculated offline for conditions in which there is not any disturbance, and a part of the controller will be set online to deal with existing disturbances or violation of $x(k+N) \in X_f$.

3.1. Offline Controller

In this section, the stability of the system will be represented when $x(k) \in X_f$ through finding control gain K . After calculating the offline controller, a procedure to specify the terminal region is defined.

Theorem 1. Taking a disturbance-free state-space model of Equation (3), there exists a control law $u = -Kx$ that asymptotically stabilizes the system if V_T is a positive definite Lyapunov function such that

- $V_T(x(k+1)) - V_T(x(k)) < 0$ for $\forall x \in X_f$ And $\forall p \in P$.
- If $x(k) \in X_f$ then $x(k+1) \in X_f$
- $|u| = |Kx| \leq u_{max}$ for $\forall x \in X_f$.

Proof. Considering $V_T = x(k)^T Q_p x(k) > 0$ as the candidate Lyapunov function, the controller $u = -Kx$ exists, if the equation below satisfies the following:

$$\begin{aligned} V_T(x(k+1)) - V_T(x(k)) &< 0 \\ x(k+1)^T Q_p x(k+1) - x(k)^T Q_p x(k) &\leq -x(k)^T Q x(k) - \Delta u(k)^T R \Delta u(k) \end{aligned} \quad (10)$$

It is assumed that there exists a K such that $A - BK$ is stable for all possible pairs of $[A(k) \ B(k)] \in \Omega$. By substituting $\Delta u = -Kx$ and $x(k+1) = (A - BK)x(k)$, it yields

$$((A - BK)x(k))^T Q_p ((A - BK)x(k)) - x(k)^T Q_p x(k) \leq -x(k)^T Q x(k) - (-Kx(k))^T R (-Kx(k)) \quad (11)$$

It can be rewritten in a more compact form:

$$M^T Q_p M - Q_p + Q + K^T R K \leq 0 \quad M = A - BK \quad (12)$$

Therefore, V_T is a Lyapunov function, and the stabilizing state-feedback gain K can be found from (12). After showing the controller's stability, the terminal region is required to be specified, reducing the conservatism level. The terminal region X_f is considered to be an ellipsoidal invariant set:

$$X_f = \left\{ x \in R^n \mid x(k+N)^T Q_p x(k+N) < \theta \right\} \quad (13)$$

While X_f is intended to be as broad as possible to reduce conservatism, it might lead to extensive control law and violation of input constraints. The maximum value of γ can be derived from an optimization problem such that input limitations are met:

$$\begin{aligned} &\max \theta \\ &\text{Subject to} \\ &x(k+N)^T Q_p x(k+N) < \theta \\ &|Kx| \leq u_{max} \end{aligned} \quad (14)$$

In Reference [10], it is proved that γ can equivalently be derived from the following optimization problem:

$$\begin{aligned} &\min \theta \\ &\theta^2 A_T P^{-1} A_T^T \leq B_T B_T^T \end{aligned} \quad (15)$$

where $A_T = [-I_n \ K \ I_n - K]^T$, $B_T = [x_{max} - x_s \ \Delta u_{max} \ x_{max} - x_s \ \Delta u_{max}]^T$, and x_s is the desired state. In this section, an offline state-feedback controller (K) and the corresponding terminal region (X_f) that stabilize the system are introduced. Unlike [17], where the future profile of scheduling variables $\{p(k+1), \dots, p(k+N)\}$ is assumed to be unknown, in this paper, a least-square algorithm was used to find scheduling variables over the prediction horizon. This means that the controller must not be robust against the possible uncertainties in the scheduling variables. For further information about the procedure for predicting the scheduling variables, please refer to [29]. \square

3.2. Online Controller

The free control moves will be determined online in the case wherein there is a disturbance or the condition $x(k) \in X_f$ is not met. The online controller steers the states towards X_f , where offline control asymptotically stabilizes the system. Two theorems are defined here to demonstrate the controller validity. Theorem 2 shows that the system remains ISpS when there is a bounded disturbance. Subsequently, the recursive feasibility of the system will be proved in Theorem 3. Because of additive disturbances, a min-max

(worst-case) optimization problem is defined to cope with uncertainties. The optimization problem (8) can be rewritten considering the bounded disturbance:

$$\begin{aligned}
 & \min_{c(k)} \max_{d(k) \in D} V_N(x(k), u(k), p(k), K, c(k)) \\
 & y(k+j) = -\left(\sum_{i=1}^{n_a} A_i q^{-i}\right) y(k+j) - \left(\sum_{j=1}^{n_b} B_j q^{-j}\right) u(k+j) + d(k), j = 1, 2, \dots, N-1 \\
 & u(k+j) = -Kx(k+j) + c(k+j) \\
 & u(k+j) \in U \\
 & \Delta u(k+j) \in V \\
 & y(k+j) \in Y \\
 & d(k+j) \in D \\
 & x(k+N) \in X_f
 \end{aligned} \tag{16}$$

The optimal solution of problem (8), subjected to the system (1) with constraints (3), is the sequence $[c^*(k) \ c^*(k+1) \ \dots \ c^*(k+N-1)]$ corresponding to $[u^*(k) \ u^*(k+1) \ \dots \ u^*(k+N-1)]$. Two theorems are defined in this section; Theorem 2 for showing stability and Theorem 3 for showing recursive feasibility. To begin with, a robust positively invariant (RPI) set is needed to be defined as bellow:

A set X is RPI for $x^+ = Ax + Bu + d$ with $u = -Kx$ if $x \in X$ for $\forall x^+ \in X, d \in D$, $[A(k) \ B(k)] \in \Omega$

Theorem 2. *There exists a sequence of optimal control input (c^*) and $\alpha, \beta, \gamma, \lambda$ that ensures the ISpS of the system (1) with bounded disturbances (6) with the following assumptions:*

- $|u| = |Kx| \leq u_{\max}$ for $\forall x \in X_f$.
- X_f is an RPI set of the system (1) with $u = -Kx$.
- $\alpha x^\lambda \leq Px \leq \beta x^\lambda$ for $\forall x \in X_f$.
- $Px^+ - Px \leq -(Qx + Rkx) + \mu d$ for $x \in X, d \in D, [A(k) \ B(k)] \in \Omega$
- $Qx + Ru \geq \gamma x^\lambda$ for $\forall x \in X_f$

The underlying causes of these assumptions can be found in [30].

Proof. The definition of ISpS for the system (1) is

$$\begin{aligned}
 & \alpha_1(\|x\|) \leq V(x) \leq \alpha_2(\|x\|) + s_1 \\
 & V(x^+) - V(x) \leq -\alpha_3(\|x\|) + \alpha_4(\|d\|) + s_2
 \end{aligned} \tag{17}$$

where x^+ RPI set in which $x^+ = Ax + Bu + d$ with $u = -Kx$; α_1, α_2 , and α_3 are \mathcal{K}_∞ -function; α_4 is \mathcal{K} -function; and s_1 and s_2 are positive numbers such that $x \in X, x^+ \in X$, and $d(k) \in D$. It is also noteworthy that ISpS is equivalent to ISS for $s_1 = s_2 = 0$. The $V_N(x) > 0$ should first be proved to be bounded so that Equation (17) can be expressed as follows:

$$V_i(x(k+i)) = \min_{c_k} \max_{d(k) \in D} \left[x(k+i)^T Qx(k+i) + u(k+i)^T Ru(k+i) + V_{i+1}(x(k+i+1)) \right] \tag{18}$$

Equation (18) is derived from a mathematical induction technique such that the following holds:

$$\begin{aligned}
 V_0(x(k)) &= \min_{c_k} \max_{d(k) \in D} \left[x(k)^T Qx(k) + u(k)^T Ru(k) + V_1(x(k+1)) \right] \\
 V_1(x(k+1)) &= \min_{c_k} \max_{d(k) \in D} \left[x(k+1)^T Qx(k+1) + u(k+1)^T Ru(k+1) + V_2(x(k+2)) \right] \\
 &\dots \\
 V_N(x(k+N)) &= [x(k+N)^T Q_p x(k+N)]
 \end{aligned} \tag{19}$$

Equation (10), when the disturbance is regarded to be non-zero, results in the following:

$$x(k+1)^T Q_p x(k+1) - x(k)^T Q_p x(k) \leq -x(k)^T Qx(k) - u(k)^T Ru(k) + d_{\max} \tag{20}$$

By developing this equation for the next value of state variables, the following equations will be reached:

$$\begin{aligned} x(k+2)^T Q_p x(k+2) - x(k)^T Q_p x(k) &\leq -x(k+1)^T Q x(k+1) - u(k+1)^T R u(k+1) + d_{max} \\ &\vdots \\ x(k+N)^T Q_p x(k+N) - x(k)^T Q_p x(k) &\leq -\sum_{i=0}^{N-1} x(k+i)^T Q x(k+i) + u(k+i)^T R u(k+i) + d_{max} \end{aligned} \quad (21)$$

From Equations (18) and (20), and substituting $u(k) = -Kx(k)$, the following is obtained.

$$V_N(x(k)) \leq \max_{d(k) \in D} \left[x(k)^T Q x(k) - x(k)^T K R x(k) \right] + x(k+1)^T Q_p x(k+1) \quad (22)$$

where $V(x_{k+1}) \leq x(k+N)^T Q_p x(k+N)$ and, therefore,

$$V_N(x) \leq x(k)^T (Q - K R + Q_p) x(k) + d_{max} = (Q - K R + Q_p) \|x\| + d_{max} \quad (23)$$

It can be concluded that $V_N(x)$ has an upper bound and $\alpha_1 = Q - K R + Q_p$, $s_1 = d_{max}$ in the right-hand side of the first equation of (23). The difference between two sequential Lyapunov functions is

$$V(x(k+1)) - V(x(k)) \leq \max_{d(k) \in D} - \left[x(k)^T Q x(k) + u(k)^T R u(k) \right] + x(k+N)^T Q_p x(k+N) \quad (24)$$

This can be expressed by

$$V(x(k+1)) - V(x(k)) \leq x(k)^T (-Q + K R + Q_p) x(k) + d_{max} = -(Q - K R - Q_p) x + d_{max} \quad (25)$$

Accordingly, the difference of Lyapunov function is bounded such that $\alpha_3 = Q - K R - Q_p$, $s_2 = d_{max}$. It has been proven that the proposed controller is ISpS, and the optimal solution, satisfying input, output, and terminal constraints, can be derived from a min-max problem (16). In the next stage, it has been shown that the solution is feasible. \square

Theorem 3. Given system (1) with constraints (2) and bounded disturbance (6), the closed-loop system is ISpS for $x_0 \in x_N$, and the system is feasible.

Proof. Suppose that there exists a terminal region, derived from (15), and a control gain K , calculated by Equation (12); then, the system is stable and $x_0 \in x_N$. Consequently, if the optimization problem (8) is feasible at time instance k , it remains feasible at the subsequent instances. After finding the optimal K , c^* , and the corresponding u^* at time instance k , the system of Equation (3) at the next instance can be rewritten.

$$x(k+1) = A(p(k))x_k + B(p(k))u_k^* + d(k) \quad (26)$$

Moreover,

$$x(k+2) = A(p(k+1))x_1 + B(p(k+1))u_0^* + d(k+1) \quad (27)$$

In general, the sequence of system states can be described as follows:

$$x(k+i+1) = A(p(k+i))x_{k+i} + B(p(k+i))u_{k+i}^* + d(k+i) \quad (28)$$

By substituting $u(k) = -Kx(k) + c(k)$, Equation (28) can be derived:

$$x(k+i+1) = (A(p(k+i)) - B(p(k+i))K) * x_{k+i} + B(p(k+i))c_{k+i}^* + d(k+i) \quad (29)$$

The above equation for $i = N$ results in

$$x(k+N+1) = (A(p(k+N)) - B(p(k+N))) * x_{k+N} + B(p(k+N))c_{k+N}^* + d(k+N) \quad (30)$$

This implies that the optimal solution is feasible because

$$\begin{aligned} & \text{for } i = 0, X_{N-1} \in X_N \\ & \text{for } i = 1, X_{N-2} \in X_{N-1} \in X_N \\ & \text{for } i = N, X_0 \in X_N \text{ and } x(k+N+1) \in X_N \end{aligned} \quad (31)$$

According to Theorems 1 to 3, the developed controller has two components. The first is determined offline, which included a control gain K and terminal region X_f . When $x \in X_f$, the offline controller asymptotically stabilizes the system. Otherwise, when there is uncertainty, an online controller verifies the ISpS. The online optimization problem is shown to be feasible. In the next section, a dynamic neural network is constructed to solve the online optimization problem for the first time. \square

4. Real-Time Optimization Problem Using RNN

The global convergence and low complexity of RNN for optimization of linear models, constrained linear models, linear models with uncertainty, and various nonlinear models are represented in the literature. In this study, RNN optimizes a real-time QP problem enjoying parallel computation. To begin with, the original optimization problem is required to transform into a standard form. The input-state relationship of Equation (1) can be expressed in the following form:

$$x(k+1) = A(p(k))x(k) + B(p(k))u(k) + d(p(k)) \quad (32)$$

The vectors of predicted outputs, inputs, and disturbance are as follows:

$$\begin{aligned} X(k) &= [X(k+1) \dots X(k+N)]^T \in \mathcal{R}^{m \times N} \\ U(k) &= [u(k) \dots u(k+N-1)]^T \in \mathcal{R}^{n \times N} \\ D(k) &= [d(k+1) \dots d(k+N)]^T \in \mathcal{R}^{o \times N} \end{aligned} \quad (33)$$

where m , n , and o are the number of states, inputs, and disturbances, respectively. The predicted states can be easily shown to be in the following form:

$$\begin{aligned} X(k+j) &= G(p(k+j-1))X(k+j-1) + F(p(k+j-1))U(k+j-1) + D(k+j-1), j = 1, \dots, N \\ G(p(k)) &= [A(p(k)) \ A(p(k))^2 \dots A(p(k))^N]^T \\ F(p(k)) &= \begin{bmatrix} B(p(k)) \\ A(p(k))B(p(k)) + B(p(k)) \\ \vdots \\ A(p(k))^{N-1}B(p(k)) + \dots + A(p(k))B(p(k)) + B(p(k)) \end{bmatrix} \end{aligned} \quad (34)$$

The constraints (2) can be represented as follows:

$$\begin{aligned} -u_{\max} &\leq u(k) \leq u_{\max} \\ -\Delta u_{\max} &\leq u(k) - u(k-1) \leq \Delta u_{\max} \\ -x_{\max} &\leq Gu + F + d \leq x_{\max} \end{aligned} \quad (35)$$

The optimization problem (16) can be written as follows:

$$\begin{aligned} \min_U X^T \bar{Q} X + U^T R U &= \min_U (GX + FU + D)^T \bar{Q} (GX + FU + D) + U^T R U \\ \min_U U^T \times ((FU)^T \bar{Q} (FU) + R) \times U &+ (GX + FU)^T \bar{Q} (GX + FU) \end{aligned} \quad (36)$$

where

$$\bar{Q} = \begin{bmatrix} Q & 0 \\ 0 & Q_p \end{bmatrix}$$

According to Equations (35) and (36), the standard form can be expressed as follows:

$$\begin{aligned} \min_v & \frac{1}{2} v^T H v + b^T v \\ \text{Subject to } & T v \leq q \end{aligned} \quad (37)$$

where

$$\begin{aligned} H &= 2 \times ((FU)^T \bar{Q} (FU) + R) \\ b &= 2 \times (GX + FU)^T \bar{Q} (GX + FU) \\ T &= \begin{bmatrix} I_{n \times n} \\ -I_{n \times n} \\ I_{n \times n} \\ -I_{n \times n} \\ G \\ -G \end{bmatrix}, \quad q = \begin{bmatrix} u_{max} \\ -u_{max} \\ \Delta u_{max} + u(k-1) \\ \Delta u_{max} - u(k-1) \\ X_{max} - F - d \\ -X_{max} + F + d \end{bmatrix} \end{aligned}$$

Remark 1. The vector q is unknown as d is not measurable, while it can be estimated by the difference between the measured output and model output. In Reference [31], a simple model to find unknown parameters is proposed.

Remark 2. In References [25,31], a simplified form of a dual neural network is described to ensure the lower computational burden by defining the dual form of the optimization problem (37) as follows:

State equation:

$$\frac{d\omega}{dt} = \lambda \left(-TH^{-1}T^T\omega + M \left(TH^{-1}T^T\omega - TH^{-1}c - \omega \right) + TH^{-1}c \right) \quad (38)$$

where ω is the state variable of the network, $\lambda > 0$ adjusts the convergence rate of RNN, and M is a piecewise linear function as follows:

$$M(z) = \begin{cases} q_{min} & z < q_{min} \\ z & q_{min} < z < q_{max} \\ q_{max} & z > q_{max} \end{cases} \quad (39)$$

Output equation:

$$v = H^{-1}T^T\omega - H^{-1}c \quad (40)$$

In the dual form of the optimization problem, constraints are added to the primary cost function as a penalty term. This means that, if $Tv < q$, the cost function decreases by a factor of α ; otherwise, for $Tv > q$, the cost function is penalized. The single-layer RNN is given in Figure 1. The global convergence of the proposed network is verified in [31].

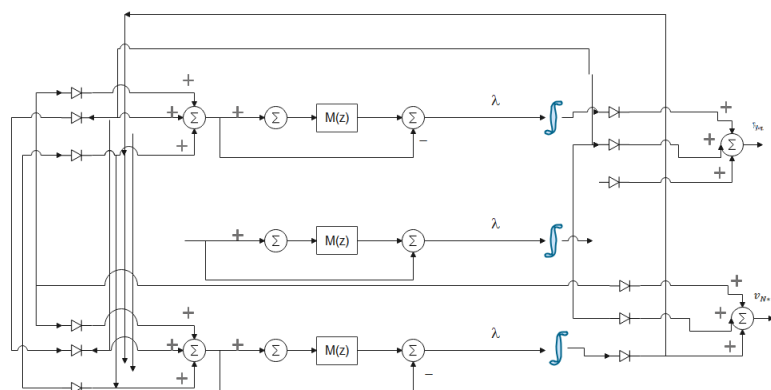


Figure 1. Recurrent neural network (RNN) framework [31].

Finally, the proposed MPC algorithm can be summarized in the following steps:

1. The value of $Q, R, \lambda, N, u_{max}, y_{max}, d_{max}, p$ and model (A, B) ;
2. Specify K using (12) and θ using (15);
3. Repeat the procedure of finding $u(k)$ by solving Equations (38)–(40).

The whole procedure is also depicted in Figure 2.

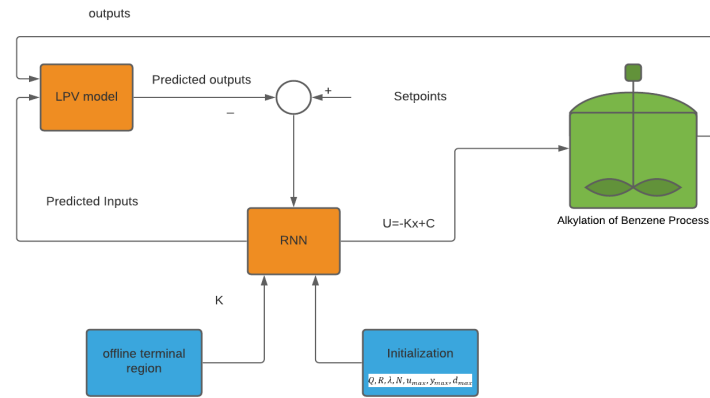


Figure 2. The proposed approach flowchart.

5. Case Study

Alkylation of benzene with ethylene, a principal process in the petrochemical industry, produces ethylbenzene, which is widely used as a large-scale benchmark. As shown in Figure 3, the process studied in this work consists of four continuously stirred tank reactors (CSTRs) and a flash tank separator. A detailed description of the process is described in [32,33]. The process's manipulated variables are heat inputs to five vessels shown by Q1, Q2, Q3, Q4, and Q5. The temperatures of five vessels T1, T2, T3, T4, and T5 are considered the process's outputs. The process's dynamic behavior is set out in detail in Equations (41)–(45), and all parameters are named in Table 1. For more information and the value of fixed parameters, refer to [34].

$$\frac{dT_1}{dt} = \frac{Q_1 + F_1 C_{A0} H_A(T_{A0}) + F_2 C_{B0} H_B(T_{B0}) + \sum_i^{A,B,C,D} F_{r2} C_{ir} H_i(T_4) - F_3 C_{i1} H_i(T_1)}{\sum_i^{A,B,C,D} C_{i1} C_{pi} V_1} + \frac{-\Delta H_{r1} r_1(T_1, C_{A1}, C_{B1}) - \Delta H_{r2} r_2(T_1, C_{B1}, C_{C1})}{\sum_i^{A,B,C,D} C_{i1} C_{pi}} \quad (41)$$

$$\frac{dT_2}{dt} = \frac{Q_2 + F_4 C_{B0} H_B(T_{B0}) + \sum_i^{A,B,C,D} F_3 C_{i1} H_i(T_1) - F_5 C_{i2} H_i(T_2)}{\sum_i^{A,B,C,D} C_{i2} C_{pi} V_2} + \frac{-\Delta H_{r1} r_1(T_2, C_{A2}, C_{B2}) - \Delta H_{r2} r_2(T_2, C_{A2}, C_{B2})}{\sum_i^{A,B,C,D} C_{i2} C_{pi}} \quad (42)$$

$$\frac{dT_3}{dt} = \frac{Q_3 + F_6 C_{B0} H_B(T_{B0}) + \sum_i^{A,B,C,D} F_5 C_{i2} H_i(T_2) - F_7 C_{i3} H_i(T_3)}{\sum_i^{A,B,C,D} C_{i3} C_{pi} V_3} + \frac{-\Delta H_{r3} r_1(T_3, C_{A3}, C_{B3}) - \Delta H_{r3} r_2(T_3, C_{B3}, C_{C3})}{\sum_i^{A,B,C,D} C_{i3} C_{pi}} \quad (43)$$

$$\frac{dT_4}{dt} = \frac{Q_4 + \sum_i^{A,B,C,D} F_7 C_{i3} H_i(T_3) + F_9 C_{i5} H_i(T_5)}{\sum_i^{A,B,C,D} C_{i4} C_{pi} V_4} + \frac{\sum_i^{A,B,C,D} -M_i H_i(T_4) - F_8 C_{i4} H_i(T_4) - M_i H_{vap_i}}{\sum_i^{A,B,C,D} C_{i4} C_{pi} V_4} \quad (44)$$

$$\frac{dT_5}{dt} = \frac{Q_5 + F_{10} C_{D0} H_D(T_{D0}) + \sum_i^{A,B,C,D} F_{r1} C_{ir} H_i(T_4) - F_9 C_{i5} H_i(T_5)}{\sum_i^{A,B,C,D} C_{i5} C_{pi} V_5} + \frac{-\Delta H_{r2} r_2(T_5, C_{B5}, C_{C5}) - \Delta H_{r3} r_3(T_5, C_{A5}, C_{D5})}{\sum_i^{A,B,C,D} C_{i5} C_{pi}} \quad (45)$$

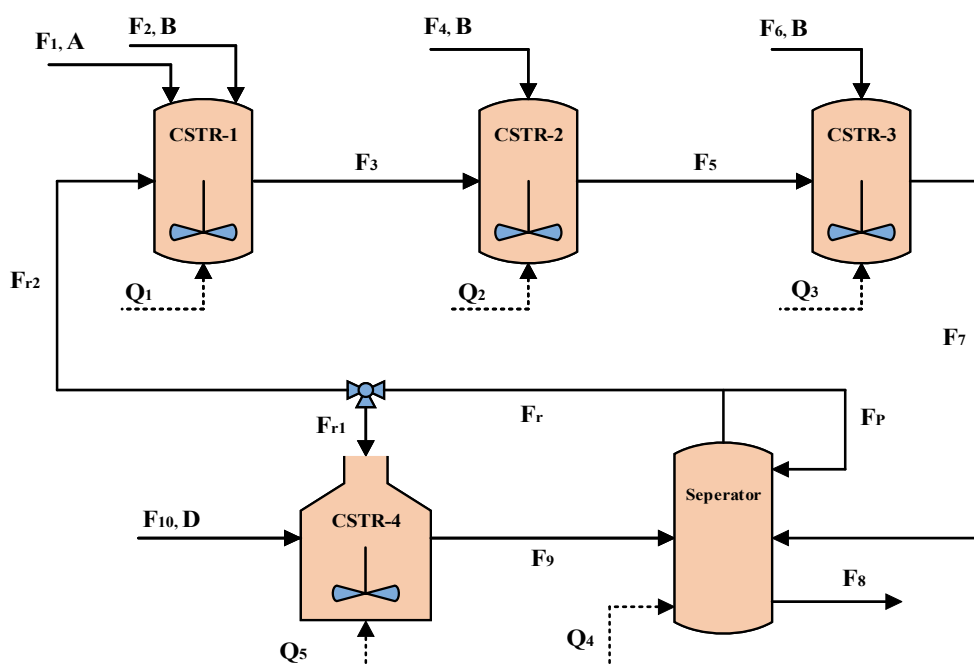


Figure 3. Flow diagram of alkylation of benzene process.

Table 1. Process variables of the alkylation of benzene process.

Variables	Definition
$C_{A1}, C_{B1}, C_{C1}, C_{D1}$	Concentrations of A, B, C, D in CSTR-1
$C_{A2}, C_{B2}, C_{C2}, C_{D2}$	Concentrations of A, B, C, D in CSTR-2
$C_{A3}, C_{B3}, C_{C3}, C_{D3}$	Concentrations of A, B, C, D in CSTR-3
$C_{A4}, C_{B4}, C_{C4}, C_{D4}$	Concentrations of A, B, C, D in Separator
$C_{A5}, C_{B5}, C_{C5}, C_{D5}$	Concentrations of A, B, C, D in CSTR-1
$C_{Ar}, C_{Br}, C_{Cr}, C_{Dr}$	Concentrations of A, B, C, D in F_r, F_{r1}, F_{r2}
T_1, T_2, T_3, T_4, T_5	Temperatures in each vessel
T_{ref}	Reference temperature
F_3, F_5, F_7, F_8, F_9	Effluent flow rates from each vessel
$F_1, F_2, F_4, F_6, F_{10}$	Feed flow rates to each vessel
F_r, F_{r1}, F_{r2}	Recycle flow rates
$H_{vapA}, H_{vapB}, H_{vapC}, H_{vapD}$	Enthalpies of vaporization of A, B, C, D
$H_{Aref}, H_{Bref}, H_{Cref}, H_{Dref}$	Enthalpies of A, B, C, D at T_{ref}
$\Delta H_{r1}, \Delta H_{r2}, \Delta H_{r3}$	Heat of reactions 1, 2, and 3
V_1, V_2, V_3, V_4, V_5	Volume of each vessel
Q_1, Q_2, Q_3, Q_4, Q_5	External heat/coolant inputs to each vessel
$C_{pA}, C_{pB}, C_{pC}, C_{pD}$	Heat capacity of A, B, C, D at liquid phase
$\alpha_A, \alpha_B, \alpha_C, \alpha_D$	Relative volatilities of A, B, C, D
$C_{A0}, C_{B0}, C_{C0}, C_{D0}$	Molar densities of pure A, B, C, D
T_{A0}, T_{B0}, T_{D0}	Feed temperatures of pure A, B, D
k	Fraction of overhead flow recycled to the reactors
$C_{A1}, C_{B1}, C_{C1}, C_{D1}$	Concentrations of A, B, C, D in CSTR-1
$C_{A2}, C_{B2}, C_{C2}, C_{D2}$	Concentrations of A, B, C, D in CSTR-2
$C_{A3}, C_{B3}, C_{C3}, C_{D3}$	Concentrations of A, B, C, D in CSTR-3
$C_{A4}, C_{B4}, C_{C4}, C_{D4}$	Concentrations of A, B, C, D in Separator
$C_{A5}, C_{B5}, C_{C5}, C_{D5}$	Concentrations of A, B, C, D in CSTR-1
$C_{Ar}, C_{Br}, C_{Cr}, C_{Dr}$	Concentrations of A, B, C, D in F_r, F_{r1}, F_{r2}
T_1, T_2, T_3, T_4, T_5	Temperatures in each vessel
T_{ref}	Reference temperature

Table 1. Cont.

Variables	Definition
F_3, F_5, F_7, F_8, F_9	Effluent flow rates from each vessel
$F_1, F_2, F_4, F_6, F_{10}$	Feed flow rates to each vessel
F_r, F_{r1}, F_{r2}	Recycle flow rates
$H_{vapA}, H_{vapB}, H_{vapC}, H_{vapD}$	Enthalpies of vaporization of A, B, C, D
$H_{Aref}, H_{Bref}, H_{Cref}, H_{Dref}$	Enthalpies of A, B, C, D at T_{ref}
$\Delta H_{r1}, \Delta H_{r2}, \Delta H_{r3}$	Heat of reactions 1, 2, and 3
V_1, V_2, V_3, V_4, V_5	Volume of each vessel
Q_1, Q_2, Q_3, Q_4, Q_5	External heat/coolant inputs to each vessel
$C_{pA}, C_{pB}, C_{pC}, C_{pD}$	Heat capacity of A, B, C, D at liquid phase
$\alpha_A, \alpha_B, \alpha_C, \alpha_D$	Relative volatilities of A, B, C, D
$C_{A0}, C_{B0}, C_{C0}, C_{D0}$	Molar densities of pure A, B, C, D

The steady-state values of inputs and outputs and initial conditions are indicated in Table 2. The controller desires to steer the system from this initial condition to the steady-states condition while satisfying the constraints:

$$|Q_1 - Q_{1s}| < 7.5 \times 10^5, |Q_2 - Q_{2s}| < 5 \times 10^5, |Q_3 - Q_{3s}| < 5 \times 10^5 \\ |Q_4 - Q_{4s}| < 6 \times 10^5, |Q_5 - Q_{5s}| < 5 \times 10^5$$

Table 2. The steady-state and initial values.

Steady-State Temperatures of Vessels (K)	Steady-State Inputs (J/K)	Initial Temperatures of Vessels (K)
$T_{1s} = 477.24$	$Q_{1s} = -4.4 \times 10^6$	$T_{1,0} = 443.02$
$T_{2s} = 476.97$	$Q_{2s} = -4.6 \times 10^6$	$T_{2,0} = 437.12$
$T_{3s} = 473.47$	$Q_{3s} = -4.7 \times 10^6$	$T_{3,0} = 428.37$
$T_{4s} = 470.60$	$Q_{4s} = 9.2 \times 10^6$	$T_{4,0} = 433.15$
$T_{5s} = 478.28$	$Q_{5s} = 5.9 \times 10^6$	$T_{5,0} = 457.55$

In contrast to previous works that developed nonlinear MPC and/or noncentralized MPC, an MPC setup built on the LPV model, improved by RNN, is proposed in this paper, having lower complexity and better performance. More significantly, F_r , C_{C0} , and C_{D0} are considered to be time-varying, for the first time, to the best of the authors' knowledge. From a practical point of view, these variables are selected to be scheduling variables with $\pm 10\%$ variation around their nominal values. Before applying the controller, the LPV-IO framework is required to be identified so that ten levels are defined within the range of 90% to 110% of the nominal value for each scheduling variable. The global LPV model originates from a polynomial interpolation of ten local LTI models, identified with 500 data points in each level (operating condition). Each local model has five inputs and five outputs. All manipulated inputs are chosen to be pseudorandom binary signals to extract data with a sampling time of $T = 10$ s. With the measured data, the LTI models, afterward, are identified using MATLAB system identification Toolbox. With several simulations, it has been realized that a third-degree polynomial interpolation produced a good fit for constructing the global LPV-IO model, while the order of local transfer functions varies from second to fifth.

The scheduling variables and the responses that come from the global LPV-IO model and nonlinear model are presented in Figures 4 and 5, respectively. The figures supplied show that the identified LPV model can predict the nonlinear system's behavior. Moreover, mean square error (MSE) and Akaike's final prediction error (FPE) for the global model were 0.032 and 0.058 in succession, and for the local LPV models, the range of MSE was between 0.004 and 0.062, and FPE changed from 0.040 to 0.081. Figure 5 and the numerical results show that the LPV-IO model is sufficiently reliable for estimating responses.

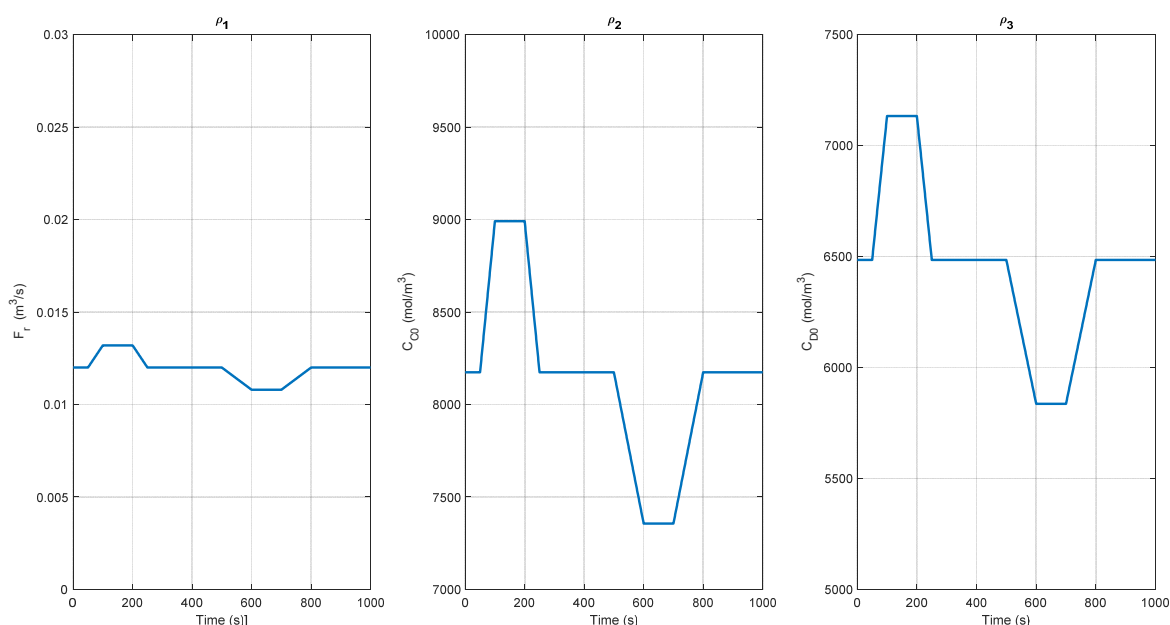


Figure 4. Scheduling variables from left to right: F_r (recycle flow rates), C_{C0} (molar densities of pure C), and C_{D0} (molar densities of pure D).

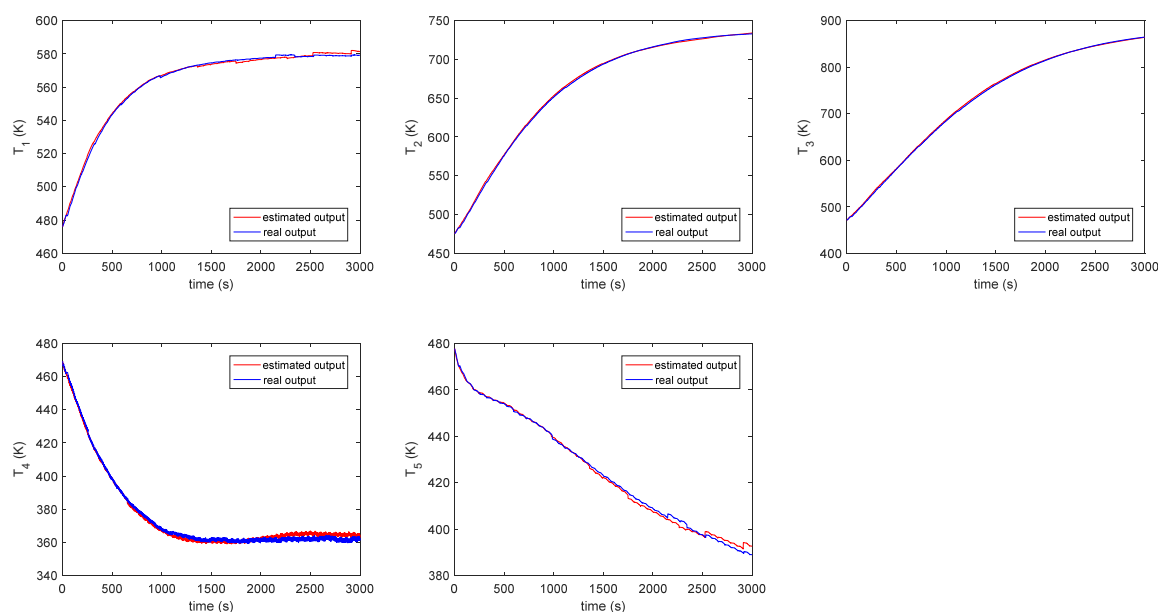


Figure 5. Temperatures in five vessels: blue and red lines indicate the measured outputs and simulated outputs from the LPV model, respectively.

6. Results and Discussion

In the next step, the performance of the controller in terms of setpoint tracking and disturbance rejection is investigated, in which a step change at $t = 1500$ s with amplitude 2×10^{-3} is made in F_1 and F_2 . The controller parameters were $N = 6$, $Q = 1000 \times I_{5 \times 5}$, $R = 10^{-7} \times I_{5 \times 5}$, and $\lambda = 3.5$. All controllers employed in the performed comparison used the same parameters. The proposed approach is compared with linear RMPC [35], and the LPV controller studied in [17] is shown in Figure 6, where the dashed red line denotes the references, the blue line is the proposed method, the green line is linear RMPC, and the black line is LPV-IO RMPC. In all methods, MPC parameters are chosen according to [36,37].

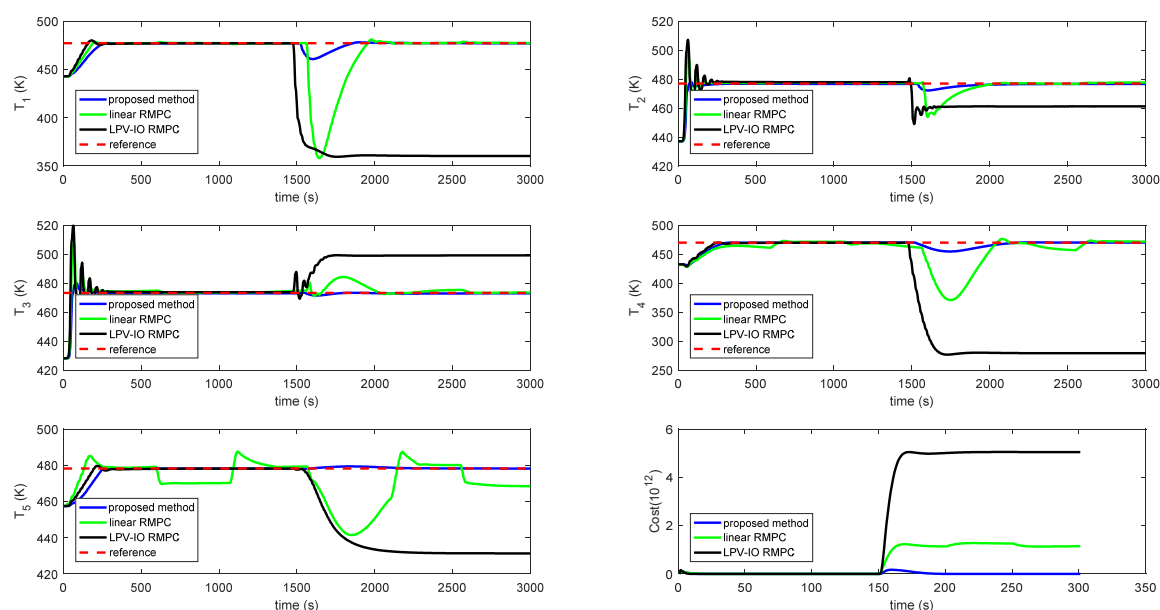


Figure 6. The temperatures of vessels and cost function using three studied methods (dashed red line: references, the blue line: proposed method, the green line: linear RMPC, the black line: LPV-IO RMPC).

Looking firstly at reference tracking, the proposed controller's rise time is slightly larger than other methods, while being fast enough. On the other hand, the proposed controller has a lower settling time. In contrast to LPV-MPCs that reach the reference, the linear RMPC has a non-zero steady-state error, especially in T_4 and T_5 . The proposed controller reached the setpoint without any oscillations, while linear RMPC and LPV-RMPC suffer from large-amplitude oscillations. The reason behind the fluctuation in T_5 for linear RMPC can be found in Equation (35), such that C_{D0} directly affects the output, and LRMPC failed to tackle the changes in molar densities of pure D. In general, regardless of rise time, settling time, and overshoot, all three methods followed the setpoints for T_1 to T_4 .

Turning to disturbance rejection, the studied approach outperforms others with removing the disturbance in a short time ranging from 120 s to 355 s for T_1 to T_5 . After time $t = 1500$ s, when the disturbance was applied to the process, the LRMPC violated both inputs and outputs constraints, especially in T_1 , although it can overcome the disturbances after a while; on the other hand, LPV-RMPC became highly unstable. Therefore, LPV-RMPC failed to cope with disturbances imposing the process in contrast to other methods, and high amplitude of oscillation in the LRMPC response can be devastating.

Regarding the cost function, the wasted resources by LPV-RMPC are 5.04 times more than LRMPC and 1.59×10 times greater than those of the proposed method, which proved that the studied approach is significantly cost-effective. More importantly, the cost function of LRMPC and LPV-RMPC did not converge to zero, so that they cannot deal with changes in three scheduling variables. The MSE of all three methods for different outputs is reported in Table 3. Using the proposed method leads to a sharp decline in error compared with other methods, as the MSE for LRMPC, LPV-RMPC, and proposed method were 447.56, 5.09×10^3 , and 43.84, respectively. In short, the proposed method had an acceptable speed (rise time, settling time, and time required to remove disturbance), as well as lower MSE and cost.

Table 3. The steady-state and initial values.

Method	T_1	T_2	T_3	T_4	T_5	Total
LRMPC	391.40	387.25	346.72	331.76	408.73	347.56
LPV-RMPC	5.33×10^3	5.31×10^3	5.07×10^3	4.90×10^3	5.41×10^3	5.09×10^3
Proposed method	67.32	65.30	52.34	60.00	76.45	43.84

After evaluating the performance of the proposed controller, the importance of RNN needs to be emphasized. Finding the global minimum in a shorter time was the primary reason behind using RNN in this study. To this end, RNN is compared with three widely used optimization algorithms for MPCs, namely, sequential quadratic programming (SQP), genetic algorithm (GA), and singular value decomposition (SVD). It has been proven that using RNN reduces the average time required for computing the control action in each sampling in contrast with NMPC based on SQP, GA, and SVD. The average run-time and MSE of these methods is reported in Table 4 and Figures 7–9. The RNN-based method experienced far less MSE (43.84) and cost (2.03×10^4) than other optimization algorithms and found the optimal control actions remarkably quick (0.033). GA ranked second in cost and MSE, while being slow, resulting in instability and significant fluctuations in some simulations. In stark contrast, however, SVD has faster responses, although it failed to converge the minimum cost function at times and saw monumental errors. SQP, on the other side, had a smaller MSE and cost than SVD and solved the optimization problem in a shorter time than GA did. According to Figure 7, RNN and SVD have the lowest and highest MSE, respectively. Figure 8 depicts that SQP and GA are sluggish as opposed to RNN and SVD. As described in Figure 9, SVD is the only method with a significant error and fails to find the proper control actions from the optimization problem.

Table 4. A comparison of different optimization algorithms for finding the control signals.

Optimization Algorithm	MSE	Average Time	Cost
RNN	43.84	0.033	2.03×10^4
SQP	153.12	0.76	8.04×10^6
GA	76.55	0.81	5.01×10^4
SVD	241.22	0.013	2.05×10^8

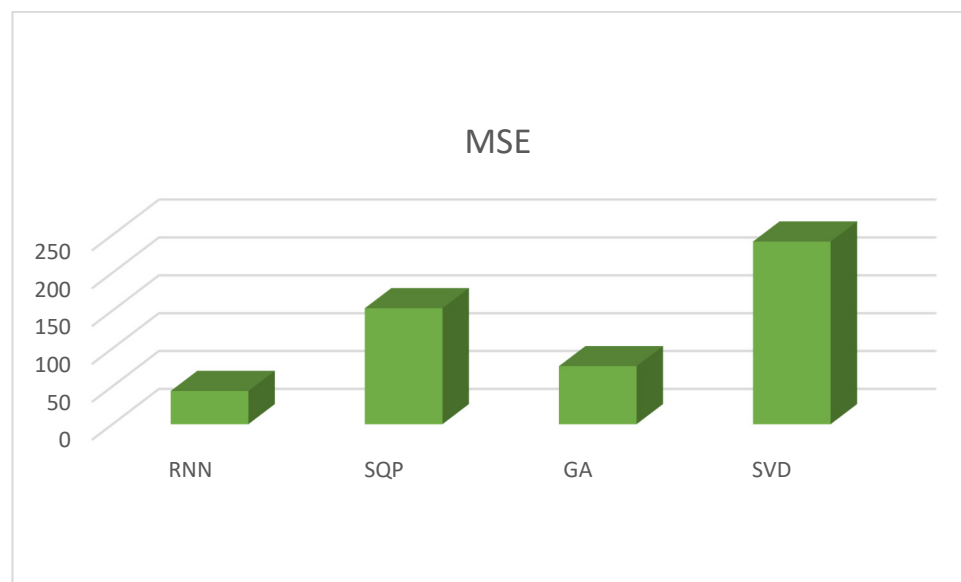


Figure 7. MSE of different optimization algorithms.

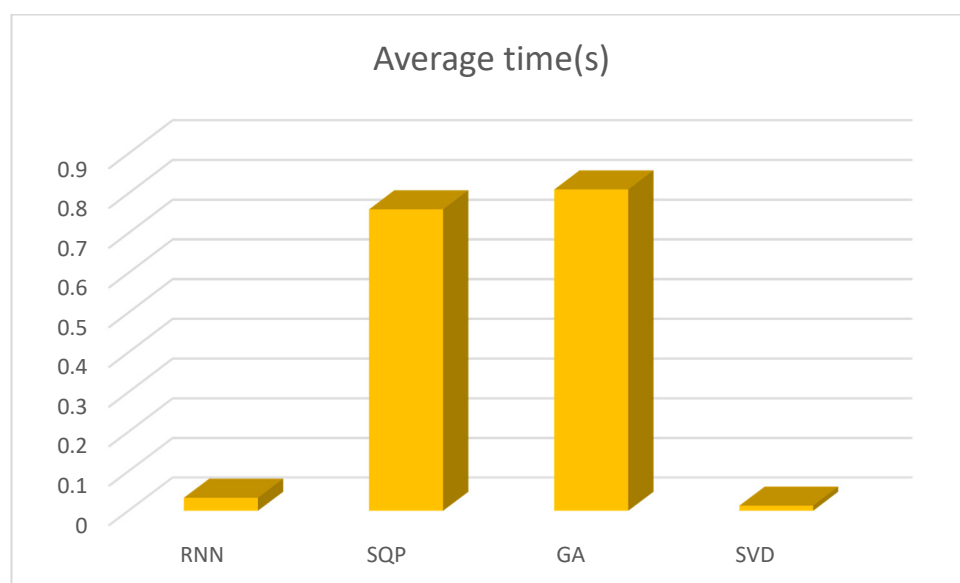


Figure 8. Average run-time of different optimization algorithms.

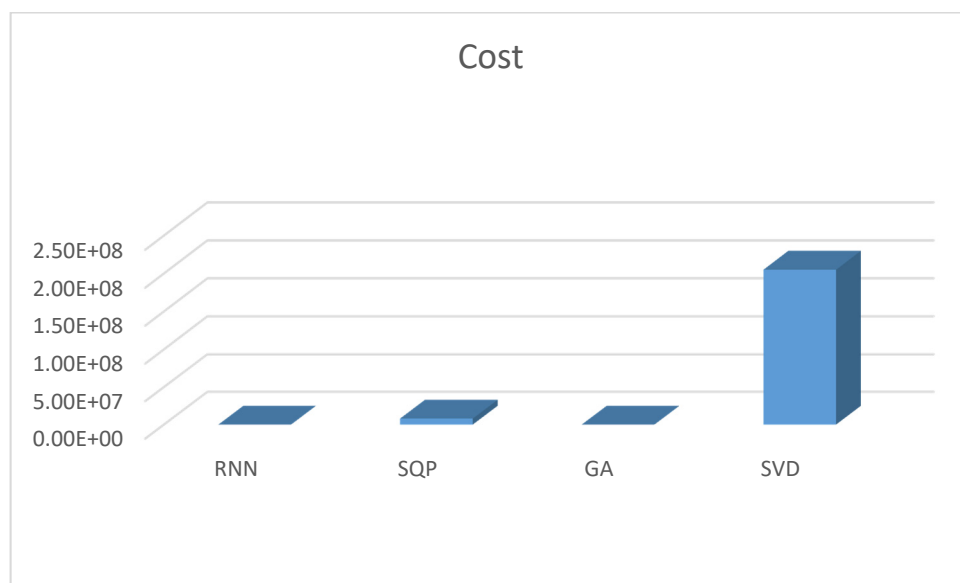


Figure 9. The cost function of different optimization algorithms.

7. Conclusions

An RMPC with an LPV-IO model is investigated in this paper, in which an RNN algorithm solves the real-time optimization problem. The study had three contributions. Firstly, an RNN-based optimization algorithm was developed to offer global convergence and lower the online computational load. Secondly, free control moves were added to the constant control gain to maintain the closed-loop stability when facing bounded disturbances. Lastly, concerning previous studies for MPC with LPV, the proposed method inherently enjoys a shrunken conservatism degree owing to finding the larger possible terminal region, using free control moves, and the global solution of the optimization problem.

This approach's effectiveness was monitored and compared with LRMPC [35] and LPV-RMPC [17] in an alkylation of benzene process, nonlinear, and large-scale with three scheduling variables. The proposed method was astonishingly successful in both setpoint tracking and disturbance rejection having a reasonable rise time, settling time, MSE, and response amplitude. The LRMPC had a similar speed response while suffering from adverse oscillation. The LPV-RMPC can partially track the predefined reference outputs,

whereas it failed to remain stable when facing disturbances. Meanwhile, four optimization algorithms are utilized to solve the online optimization problem of the proposed controller. The results showed that RNN and GA outdid in error reduction and finding the optimal solution; nonetheless, GA was sluggish, and thus detrimental to stability. SVD had the fastest convergence rate with the highest MSE and cost value, and SQP, with a stable response, had the worst execution with an excellent average time and MSE. Future works will focus on showing the stability of the LPV-IO system with multiplicative disturbances. Moreover, the conservatism can be further lowered by ellipsoidal invariant sets with polyhedral invariant sets.

Author Contributions: Conceptualization, M.H., A.R. and W.Z.; methodology, M.H.; software, M.H.; validation, M.H., A.R. and W.Z.; formal analysis, M.H.; investigation, M.H.; resources, M.H., A.R. and W.Z.; writing—original draft preparation, M.H.; writing—review and editing, M.H., A.R. and W.Z.; visualization, M.H.; supervision, A.R. and W.Z.; project administration, A.R. and W.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

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

References

- Camacho, E.F.; Alba, C.B. *Model Predictive Control*; Springer Science & Business Media: Berlin, Germany, 2013.
- Yamasu, V.; Wu, B. *Model Predictive Control of Wind Energy Conversion Systems*; John Wiley & Sons: Hoboken, NJ, USA, 2016.
- Raković, S.V.; Levine, W.S. *Handbook of Model Predictive Control*; Springer: Berlin/Heidelberg, Germany, 2018.
- Zhang, R.; Xue, A.; Gao, F. *Model Predictive Control*; Springer Science and Business Media LLC: Berlin, Germany, 2019.
- Morato, M.M.; Normey-Rico, J.E.; Senname, O. Model predictive control design for linear parameter varying systems: A survey. *Annu. Rev. Control.* **2020**, *49*, 64–80. [[CrossRef](#)]
- Yu, S.; Böhm, C.; Chen, H.; Allgöwer, F. Model predictive control of constrained LPV systems. *Int. J. Control* **2012**, *85*, 671–683. [[CrossRef](#)]
- Yang, Y.; Ding, B. Model predictive control for LPV models with maximal stabilizable model range. *Asian J. Control* **2020**, *22*, 1940–1950. [[CrossRef](#)]
- Hu, C.; Wei, X.; Ren, Y. Passive fault-tolerant control based on weighted LPV tube-MPC for air-breathing hypersonic vehicles. *Int. J. Control Autom. Syst.* **2019**, *17*, 1957–1970. [[CrossRef](#)]
- Wang, X.; Zhang, X.; Yang, X. Delay-dependent robust dissipative control for singular LPV systems with multiple input delays. *Int. J. Control Autom. Syst.* **2019**, *17*, 327–335. [[CrossRef](#)]
- Abbas, H.S.; Toth, R.; Meskin, N.; Mohammadpour, J.; Hanema, J. A robust MPC for input-output LPV Models. *IEEE Trans. Autom. Control* **2016**, *61*, 4183–4188. [[CrossRef](#)]
- Ding, B.; Wang, P.; Hu, J. Dynamic output feedback robust MPC with one free control move for LPV model with bounded disturbance. *Asian J. Control* **2017**, *20*, 755–767. [[CrossRef](#)]
- Hu, J.; Ding, B. One-step ahead robust MPC for LPV model with bounded disturbance. *Eur. J. Control* **2020**, *52*, 59–66. [[CrossRef](#)]
- Alcalá, E.; Puig, V.; Quevedo, J.; Rosolia, U. Autonomous racing using linear parameter varying-model predictive control (LPV-MPC). *Control Eng. Pract.* **2020**, *95*, 104270. [[CrossRef](#)]
- Li, D.; Xi, Y. The feedback robust MPC for LPV systems with bounded rates of parameter changes. *IEEE Trans. Autom. Control* **2010**, *55*, 503–507.
- Xu, Z.; Zhao, J.; Qian, J.; Zhu, Y. Nonlinear MPC using an identified LPV model. *Ind. Eng. Chem. Res.* **2009**, *48*, 3043–3051. [[CrossRef](#)]
- Calderón, H.M.; Cisneros, P.S.; Werner, H. qLPV predictive control-A benchmark study on state space vs input-output approach. *IFAC-Pap.* **2019**, *52*, 146–151. [[CrossRef](#)]
- Abbas, H.S.; Hanema, J.; Tóth, R.; Mohammadpour, J.; Meskin, N. An improved robust model predictive control for linear parameter-varying input-output models. *Int. J. Robust Nonlinear Control* **2018**, *28*, 859–880. [[CrossRef](#)]
- Cisneros, P.S.G.; Werner, H. Stabilizing model predictive control for nonlinear systems in input-output quasi-LPV form. In Proceedings of the 2019 American Control Conference (ACC), Philadelphia, PA, USA, 10–12 July 2019; pp. 1002–1007.
- Bouzerdoum, A.; Pattison, T. Neural network for quadratic optimization with bound constraints. *IEEE Trans. Neural Netw.* **1993**, *4*, 293–304. [[CrossRef](#)]
- Tan, K.C.; Tang, H.; Yi, Z. Global exponential stability of discrete-time neural networks for constrained quadratic optimization. *Neurocomputing* **2004**, *56*, 399–406. [[CrossRef](#)]
- Xia, Y.; Wang, J. A recurrent neural network for nonlinear convex optimization subject to nonlinear inequality constraints. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2004**, *51*, 1385–1394. [[CrossRef](#)]

22. Xia, Y.; Feng, G.; Wang, J. A novel recurrent neural network for solving nonlinear optimization problems with inequality constraints. *IEEE Trans. Neural Netw.* **2008**, *19*, 1340–1353. [[PubMed](#)]
23. Xia, Y.; Wang, J. A general projection neural network for solving monotone variational inequalities and related optimization problems. *IEEE Trans. Neural Netw.* **2004**, *15*, 318–328. [[CrossRef](#)] [[PubMed](#)]
24. Nguyen, L.V.; Qin, X. Some Results on Strongly Pseudomonotone Quasi-Variational Inequalities. *Set-Valued Var. Anal.* **2019**, *28*, 239–257. [[CrossRef](#)]
25. Pan, Y.; Wang, J. Model predictive control for nonlinear affine systems based on the simplified dual neural network. In Proceedings of the 2009 IEEE International Conference on Control Applications, St. Petersburg, Russia, 8–10 July 2009; pp. 683–688.
26. Wu, Z.; Christofides, P.D. Optimizing process economics and operational safety via economic MPC using barrier functions and recurrent neural network models. *Chem. Eng. Res. Des.* **2019**, *152*, 455–465. [[CrossRef](#)]
27. Lanzetti, N.; Lian, Y.Z.; Cortinovis, A.; Dominguez, L.; Mercangoz, M.; Jones, C. Recurrent neural network based MPC for process industries. In Proceedings of the 2019 18th European Control Conference (ECC), Naples, Italy, 25–28 June 2019; pp. 1005–1010.
28. Wollnack, S.; Werner, H. LPV-IO controller design: An LMI approach. In Proceedings of the 2016 American Control Conference (ACC); Institute of Electrical and Electronics Engineers (IEEE), Boston, MA, USA, 6–8 July 2016; pp. 4617–4622.
29. Morato, M.M.; Normey-Rico, J.E.; Sename, O. Novel qLPV MPC Design with Least-Squares Scheduling Prediction. *IFAC Pap.* **2019**, *52*, 158–163. [[CrossRef](#)]
30. Zhang, S.; Dai, L.; Xia, Y. Adaptive MPC for constrained systems with parameter uncertainty and additive disturbance. *IET Control Theory Appl.* **2019**, *13*, 2500–2506. [[CrossRef](#)]
31. Pan, Y.; Wang, J. A neurodynamic optimization approach to nonlinear model predictive control. In Proceedings of the 2010 IEEE International Conference on Systems, Man and Cybernetics; Institute of Electrical and Electronics Engineers (IEEE), Istanbul, Turkey, 10–13 October 2010; pp. 1597–1602.
32. Salahshoor, K.; Hadian, M. A decentralized event-based model predictive controller design method for large-scale systems. *Autom. Control. Inf. Sci.* **2014**, *2*, 26–31.
33. Liu, J.; Chen, X.; De La Pena, D.M.; Christofides, P.D. Sequential and iterative architectures for distributed model predictive control of nonlinear process systems. Part I: Theory. *AIChE J.* **2010**, *56*, 3148–3155.
34. Kayacan, E.; Ramon, H.; Saeys, W. Distributed nonlinear model predictive control of an autonomous tractor–trailer system. *Mechatronics* **2014**, *24*, 926–933. [[CrossRef](#)]
35. Hu, J.; Ding, B. Output feedback robust MPC for linear systems with norm-bounded model uncertainty and disturbance. *Automatica* **2019**, *108*, 108489. [[CrossRef](#)]
36. Moradmand, A.; Dorostian, M.; Shafai, B. Energy scheduling for residential distributed energy resources with uncertainties using model-based predictive control. *JEPE* **2021**, *132*, 107074.
37. Dorostian, M.; Moradmand, A. Hierarchical Robust Model-based Predictive Control in Supply Chain Management under Demand Uncertainty and Time-delay. In Proceedings of the 2021 7th International Conference on Control, Instrumentation and Automation (ICCIA), Tabriz, Iran, 23–24 February 2021; pp. 1–6.