



# Article Multi-Parameter Quadratic Programming Explicit Model Predictive Based Real Time Turboshaft Engine Control

Nannan Gu<sup>1</sup>, Xi Wang<sup>1</sup> and Meiyin Zhu<sup>2,\*</sup>

- <sup>1</sup> School of Energy and Power Engineering, Beihang University, Beijing 100191, China; by1404101@buaa.edu.cn (N.G.); xwang@buaa.edu.cn (X.W.)
- <sup>2</sup> Beihang Hangzhou Innovation Institute Yuhang, Beihang University, Hangzhou 310023, China
- \* Correspondence: mecalzmy@buaa.edu.cn; Tel.: +86-156-5037-4765

Abstract: The traditional model predictive control (tMPC) algorithms have a large amount of online calculation, which makes it difficult to apply them directly to turboshaft engine-rotor systems because of real time requirements. Therefore, based on the theory of the perturbed piecewise affine system (PWA) and multi-parameter quadratic programming explicit model predictive control (mpQP-eMPC) algorithm, we develop a controller design method for turboshaft engine-rotor systems, which can be used for engine steady-state, transient state and limit protection control. This method consists of two steps: controller offline design and online implementation. Firstly, the parameter space of the PWA system is divided into several partitions offline based on the disturbance and performance constraints. Each partition has its own control law, which is in the form of piecewise affine linear function between the controller and the parameters. The control laws for those partitions are also obtained in this offline step. After which, for the online control implementation step, the corresponding control law can be obtained by a real-time query of a corresponding partition, which the current engine state falls into. This greatly reduces the amount of online calculation and thus improves the real-time performance of the MPC controller. The effectiveness of the proposed method is verified by simulating the steadystate and transient process of a turboshaft engine-rotor system with a limit protection requirement. Compared with tMPC, an mpQP-eMPC based controller can not only guarantee good steady-state, dynamic control performance and limit protection, but can also significantly improve the real-time performance of the control system.

**Keywords:** real-time control; turboshaft engine limit protection; perturbed piecewise affine systems; multi-parameter quadratic programming; explicit model predictive control

## 1. Introduction

Helicopters play an important role in short distance transportation, aerial photography, agriculture, disaster search and rescue and other fields. The power unit of a helicopter is important and the turboshaft engine is the main selection for most helicopters in the world because of its high power-to-weight ratio and low vibration. Therefore, improving the reliability and advancement of the turboshaft engine control system is the key to improving the overall maneuverability of the helicopter to meet the needs of modern industry, agriculture and military [1]. Since the dynamics of the propeller rotor and the engine are coupled through the gear transmission system, the working state of the turboshaft engine will be affected by the propeller rotor load with large inertia. The low dynamic characteristic of a high inertia rotor load creates a great challenge for the design of a high performance control system.

The difficulties in the design of turboshaft engine control systems are mainly reflected in two aspects. First, the turboshaft engine and the rotor system are directly connected through physical structures, such as gearboxes and driven shafts, which makes a strong dynamics coupling between the two independent systems. Currently, for most turboshaft



Citation: Gu, N.; Wang, X.; Zhu, M. Multi-Parameter Quadratic Programming Explicit Model Predictive Based Real Time Turboshaft Engine Control. *Energies* 2021, 14, 5539. https://doi.org/ 10.3390/en14175539

Academic Editors: Andrea Bonfiglio and Andrea Mazza

Received: 11 August 2021 Accepted: 30 August 2021 Published: 4 September 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/). engines, a traditional PID linear controller with MAX–MIN high–low selection logic structure is widely used. However, for this nonlinear dynamic system with strong coupling dynamics and strong constraints, the performance of this kind of controller is unsatisfying. The acceleration and deceleration transition is slow, and the overshoot and droop can reach 4%, which causes great damage to the engine's life span and safety [2]. Therefore, the traditional PID controller has been unable to meet the performance requirements of modern high-performance turboshaft engines due to its limitations. The research on new control algorithms to adapt to the rapid development of turboshaft engines has become of great interest to controller designers. Thanks to the development of MPC methods, research on MPC methods for turboshaft engine control has been attractive to the engineering community.

MPC originated in the 1970s. Due to its unique advantages, such as the ability to predict future output, process constrained optimization, and the ability to suppress unmodeled dynamic error and environmental disturbance uncertainty, it has been widely discussed by the industry [3–5], and has been successfully applied in industrial process control with slow dynamic characteristics such as petroleum smelting and chemical industry [6-8]. Currently, all the application research of turboshaft engines that we can find is particularly scarce and most of the model predictive controller designs for aeroengines are focused on solving the model mismatch phenomenon. At the present time, the real-time research on model predictive controllers for turboshafts is limited. However, in the face of the complex nonlinear dynamic system of turboshaft engines with strong rotor coupled dynamics and strong constraint conditions, and due to the huge amount of calculation in the optimization process, it is difficult to apply in real-time and online engineering, and its development has thus been limited in recent years. In summary, since there are few model predictive real-time controller designs of turboshaft engines and the real-time performance is of great importance for engines, mpQP-eMPC real-time research is carried out in this paper.

In order to solve the problem introduced by the large number of online calculations of tMPC, Bemporad proposed the mpQP-eMPC algorithm based on multi-parameter quadratic programming theory in [9]. Parametric programming is a kind of optimization problem with parameters; multi-parameter quadratic programming is popular in academia because of its convexity [10,11]. Due to the partition of the parameter space, the multi-parameter quadratic programming has the ability to obtain the explicit controller offline, which can simplify the process of the online solution [12–14]. The mpQP-eMPC based on parametric quadratic programming can reduce the online computing time to meet the real-time requirements. Considering the high real-time requirements of a turboshaft engine rotor system's control system, this paper will design a turboshaft engine controller based on mpQP-eMPC to solve the problem of tMPC being unable to guarantee real-time performance.

Based on the perturbed PWA model of turboshaft engine rotor systems, through theoretical derivation, this paper transforms the control problem of turboshaft engines into a standard mpQP-eMPC controller design problem. The specific solution of the affine function relationship is also derived. The control design process of turboshaft engines proposed in this paper is described as follows: Firstly, the parameter space is reasonably partitioned. By way of the derivation introduced in this paper, the detailed piecewise affine linear function relationship between the controller and the related parameters on each parameter partition is obtained. These are all conducted in the offline step. Then, during online implementation, the optimal control decision of the turboshaft engine–rotor system is obtained by querying the partition of the system state. The simulation results show that mpQP-eMPC is better than tMPC in ensuring the steady-state and transient dynamic performance of the turboshaft engine rotor system, and the average calculation time of mpQP-eMPC is 0.002 times that of tMPC, which can meet the real-time requirements.

The rest of the paper is organized as follows: Section 2 briefly introduces the turboshaft engine–rotor system and its perturbed PWA model. Section 3 presents the mpQP-eMPC

algorithm for the turboshaft engine–rotor system. In Section 4, a simulation study on a turboshaft engine–rotor system is presented to verify the proposed method. Finally, Section 5 concludes the paper.

#### 2. Turboshaft Engine–Rotor System and Its Perturbed PWA Model

The turboshaft engine–rotor system is shown in Figure 1 and its main control objective is to keep constant power turbine rotation speed by quickly adjusting the gas turbine rotation speed through changing the input fuel flow when the collective pitch changes under different flight tasks. This is necessary to meet the requirements of the rotor load changes of the helicopter. The controller should ensure that the acceleration and deceleration time is as short as possible during the transition state, and the key performance parameters of the engine are within the limits in the meantime.



Figure 1. The structure diagram of the turboshaft engine–rotor system.

The state space nonlinear equations of the turboshaft engine–rotor system are described as follows:

$$\begin{bmatrix} n_g \\ n_p \end{bmatrix} = f\left(\begin{bmatrix} n_g \\ n_p \end{bmatrix}, w_f, xcpc, H, Ma\right), \tag{1}$$

where  $\begin{bmatrix} n_g \\ n_p \end{bmatrix}$  are the state variables of the system,  $n_g$  is the rotation speed of the gas turbine,  $n_p$  is the rotation speed of the power turbine,  $w_f$  is the fuel input of the turboshaft engine, and the disturbance input is the collective pitch input *xcpc* of the rotor system, the flight altitude is *H* and the flight Mach number is *Ma*.

Equation (1) only highlights the input and output parameters without providing the explicit relationship between them. In fact, we cannot build this explicit input–output function expression of the engine. Therefore, for the requirements of controller design, the construction work of the PWA system is carried out based on the component-level model shown in Figure 1.

Since the dynamic process of the turboshaft engine–rotor system (shown in Figure 1) and Equation (1) is highly nonlinear and has strong coupling characteristics, the controller design is completed by considering the PWA modeling method. Firstly, the nonlinear state

space model of the turboshaft engine–rotor system is linearized in the ground state (H = 0, Ma = 0). At *s* steady-state working points, the discrete turboshaft engine rotor-system PWA model is established:

$$\begin{cases} x^{i}(k+1) = A^{i}x^{i}(k) + B^{i}u^{i}(k) + B^{i}_{w}w^{i}, \forall \begin{bmatrix} x^{i}(k) \\ u^{i}(k) \end{bmatrix} \in \Omega^{i}, i = 1 \dots s \\ y^{i}(k) = C^{i}x^{i}(k), \end{cases}$$
(2)

where  $x^{i} = x^{i} - x_{0}^{i} = \begin{bmatrix} n_{g}^{i} \\ n_{p}^{i} \end{bmatrix} - \begin{bmatrix} n_{g0}^{i} \\ n_{p0}^{i} \end{bmatrix}$ ,  $u^{i} = w_{f}^{i} - w_{f0}^{i}$ ,  $w^{i} = xcpc^{i} - xcpc_{0}^{i}$ , *i* represents the ith linear system. High altitude and other flight conditions can be transformed into ground

ith linear system. High altitude and other flight conditions can be transformed into ground states by similarity theory, so as to realize the full flight envelope control of the engine.

**Remark 1.** Through a lot of simulation research and a contrast between the PWA and the turboshaft engine results, the final value of s is determined with s = 10. The matching between the two when s = 10 will be proven in Section 4.1.

## 3. mpQP-eMPC Algorithm for Turboshaft Engine-Rotor System

The mpQP-eMPC structure of the turboshaft engine–rotor system is shown in Figure 2. The design process can be divided into two steps:

- The tMPC optimization problem of the turboshaft engine-rotor system is transformed into a standard mpQP-eMPC problem;
- The turboshaft engine controller design based on mpQP-eMPC.



Figure 2. The structure diagram of mpQP-eMPC.

#### 3.1. Standard mpQP-eMPC Algorithm

Firstly, the standard description of mpQP-eMPC with parameter  $z_1$  is shown as follows:

$$\min_{Z} \frac{1}{2} Z^{T} H Z$$
(3)
  
s.t.  $GZ \leq W + SZ_{1}$ ,

where *Z* is the optimization variable and  $Z_1$  is the optimization parameter. Matrices *H*, *G* and *W* are the weight matrix of objective function and constraints matrices, respectively. One of the objectives in this paper is to convert the parameters in Equation (2) and the classical MPC control into the standard mpQP-eMPC problem in Equation (3).

The mpQP-eMPC explicit solution is based on the concepts of polytopic partition and piecewise affine function.

**Definition 1** ([10]). If a polyhedron set  $\{\vartheta_l\}_{l=0}^L = \{\vartheta_0, \vartheta_1, \cdots, \vartheta_L\}$  and a polyhedron  $\vartheta$  satisfy the following conditions, then the polyhedron set  $\{\vartheta_l\}_{l=0}^L = \{\vartheta_0, \vartheta_1, \cdots, \vartheta_L\}$  is a polyhedron partition of polyhedron  $\vartheta$ :

$$\bigcup_{l=0}^{L} \vartheta_{l} = \vartheta$$

$$(\vartheta_{l}/\partial\vartheta_{l}) \bigcap (\vartheta_{q}/\partial\vartheta_{q}) = \emptyset, \forall l \neq q.$$
(4)

**Definition 2** ([10]). Consider the function  $f : \vartheta \to R^d$ ,  $d \in \mathbb{N}^+ \triangleq \{1, 2, \dots, \}$ , if there is always a polyhedral partition of the set  $\vartheta$ ,  $\{\vartheta_l\}_{l=0}^L = \{\vartheta_0, \vartheta_1, \dots, \vartheta_L\}$ , such that  $f(Z_1) = F_l Z_1 + G_l$ ,  $Z_1 \in \vartheta_l$ , then the function f is said to be piecewise affine.

Based on Definitions 1 and 2 and Lemma 1, the explicit optimal solution of (3) can be obtained.

**Lemma 1** ([9]). Based on the multi-parameter programming method to solve the optimization problems (3), then its feasible region is convex and the optimal solution is continuous and piecewise affine:

$$u(Z_1) = F_l Z_1 + G_l, Z_1 \in \vartheta_l = \left\{ Z_1 \in \mathbb{R}^d | H_l Z_1 \leqslant K_l \right\}.$$

$$(5)$$

Lemma 1 serves as a tool to obtain the explicit expression of the mpQP-eMPC controller for the turboshaft engine–rotor and will be used in the rest of this article.

#### 3.2. Transformation of Turboshaft Engine Controller and mpQP-eMPC Controller

The traditional tMPC control algorithm of the turboshaft engine is described as follows:

$$\min_{\substack{U = \{u(k), \dots, u(k+N_u-1)\}\\ s.t.}} \sum_{j=1}^{ry} \left\{ [y(k+j|k) - r(k)]^T Q[y(k+j) - r(k)] + U(k+j|k)^T R U(k+j|k) \right\}$$
(6)

 $y_{\min} \le y_{k+j|k} \le y_{\max}$  $u_{\min} \le u_{k+j} \le u_{\max}$ .

In order to transform the tMPC of the turboshaft engine into the standard mpQPeMPC form, the state prediction equation and output prediction equation of the perturbed PWA system are calculated based on (2).

State prediction equation:

$$\begin{bmatrix} x^{i}(k+1) \\ \vdots \\ x^{i}(k+n_{y}) \end{bmatrix} = P_{x}^{i}x^{i}(k) + P_{u}^{i}U^{i}(k) + P_{w}^{i}\varpi^{i}(k).$$

$$(7)$$

Output prediction equation:

$$\begin{bmatrix} y^{i}(k+1) \\ \vdots \\ y^{i}(k+n_{y}) \end{bmatrix} = \begin{bmatrix} Cx^{i}(k+1) \\ \vdots \\ Cx^{i}(k+n_{y}) \end{bmatrix} = P_{yx}^{i}x^{i}(k) + P_{yu}^{i}U^{i}(k) + P_{yw}^{i}\omega^{i}(k), \quad (8)$$

where

$$\begin{split} P_{x}^{i} &= \begin{bmatrix} A \\ \vdots \\ A^{N_{y}} \end{bmatrix}, P_{u}^{i} = \begin{bmatrix} B^{i} & 0 & \cdots & 0 \\ A^{i}B^{i} & B^{i} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N_{y}-1}B^{i} & A^{N_{y}-2}B^{i} & \cdots & B^{i} \end{bmatrix} \\ U^{i}(k) &= \begin{bmatrix} u(k) \\ \vdots \\ u(k+N_{y}-1) \end{bmatrix}, P_{w}^{i} = \begin{bmatrix} B^{i}_{w} & 0 & \cdots & 0 \\ A^{i}B^{i}_{w} & B^{i}_{w} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N_{y}-1}B^{i}_{w} & A^{N_{y}-2}B^{i}_{w} & \cdots & B^{i}_{w} \end{bmatrix} \\ & \varphi^{i}(k) &= \begin{bmatrix} w(k) \\ \vdots \\ w(k+N_{y}-1) \end{bmatrix}, P_{yx}^{i} = \begin{bmatrix} C^{i}A \\ \vdots \\ C^{i}A^{N_{y}} \end{bmatrix} \\ & P_{yw}^{i} = \begin{bmatrix} C^{i}B^{i}_{w} & 0 & \cdots & 0 \\ C^{i}A^{i}B^{i}_{w} & C^{i}B^{i}_{w} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ C^{i}A^{N_{y}-1}B^{i}_{w} & C^{i}A^{N_{y}-2}B^{i}_{w} & \cdots & C^{i}B^{i}_{w} \end{bmatrix} \\ & P_{yu}^{i} = \begin{bmatrix} C^{i}B^{i} & 0 & \cdots & 0 \\ C^{i}A^{i}B^{i} & C^{i}B^{i} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ C^{i}A^{N_{y}-1}B^{i} & C^{i}A^{N_{y}-2}B^{i} & \cdots & C^{i}B^{i}_{w} \end{bmatrix} \end{split}$$

The parameter constraints conditions of Equation (6) are equivalent to:

$$\sum_{j=0}^{N_{y}-1} \left\{ \begin{bmatrix} y^{i}(k+j|k) - r(k) \end{bmatrix}^{T} Q^{i} \begin{bmatrix} y^{i}(k+j) - r(k) \end{bmatrix} + u^{i}(k+j|k)^{T} R u^{i}(k+j|k) \right\}$$

$$= \begin{bmatrix} y^{i}(k|k) - r(k) \\ \vdots \\ y^{i}(k+n_{y}-1|k) - r(k+n_{y}-1) \end{bmatrix}^{T} Q^{i} \begin{bmatrix} y^{i}(k|k) - r(k) \\ \vdots \\ y^{i}(k+n_{y}-1|k) - r(k+n_{y}-1) \end{bmatrix} + U^{i^{T}} R^{i} U^{i}.$$
(9)
Eurthermore, the prediction Equations in (7) and (8) are brought into Equation (9)

Furthermore, the prediction Equations in (7) and (8) are brought into Equation (9) to obtain:

$$\sum_{j=0}^{N_{y}-1} \left\{ \left[ y^{i}(k+j|k) - r(k) \right]^{T} Q^{i} \left[ y^{i}(k+j) - r(k) \right] + u^{i}(k+j|k)^{T} R u^{i}(k+j|k) \right\} \\
= \left\{ P^{i}_{yx} x^{i}(k) + P^{i}_{yu} U^{i}(k) + P^{i}_{yw} \varpi^{i}(k) - \bar{r}(k) \right\}^{T} Q^{i} \left\{ P^{i}_{yx} x^{i}(k) + P^{i}_{yu} U^{i}(k) + P^{i}_{yw} \varpi^{i}(k) - \bar{r}(k) \right\} + U^{i}^{T} R^{i} U^{i} \\
= \left\{ P^{i}_{yx} x^{i}(k) + P^{i}_{yu} U^{i}(k) + P^{i}_{yw} \varpi^{i}(k) - \bar{r}(k) \right\}^{T} Q^{i} \left\{ P^{i}_{yx} x^{i}(k) + P^{i}_{yu} U^{i}(k) + P^{i}_{yw} \varpi^{i}(k) - \bar{r}(k) \right\} + U^{i}^{T} R^{i} U^{i} \\
= \left\{ \left[ P^{i}_{yx} P^{i}_{yw} - I^{i} \right] \left[ \frac{x^{i}(k)}{\varpi^{i}(k)} \right] + P^{i}_{yu} U^{i} \right\}^{T} Q \left\{ \left[ P^{i}_{yx} P^{i}_{yw} - I^{i} \right] \left[ \frac{x^{i}(k)}{\varpi^{i}(k)} \right] + P^{i}_{yu} U^{i} \right\} + U^{i}^{T} R^{i} U^{i}. \\$$
Let

$$\Pi_{1} = \begin{bmatrix} P_{yx}^{i} & P_{yw}^{i} & -I^{i} \end{bmatrix}^{T} Q^{i} \begin{bmatrix} P_{yx}^{i} & P_{yw}^{i} & -I^{i} \end{bmatrix}$$
$$\Pi_{2} = \begin{bmatrix} P_{yx}^{i} & P_{yw}^{i} & -I^{i} \end{bmatrix}^{T} Q^{i} P_{yu}^{i} U^{i}$$
$$\Pi_{3} = \begin{pmatrix} P_{yu}^{i} \end{pmatrix}^{T} Q^{i} P_{yu}^{i} + R^{i},$$

$$\begin{aligned} & \text{then:} \\
& \min_{\substack{U = \{u_k, \dots, u_{k+N_u-1}\} \\ U = \{u_k, \dots, u_{k+N_u-1}\}}} \sum_{j=0}^{N_y-1} \left\{ \left[ y^i(k+j|k) - r(k) \right]^T Q^i \left[ y^i(k+j) - r(k) \right] + u^i(k+j|k)^T R u^i(k+j|k) \right\} \\
& = \begin{bmatrix} x^i(k) \\ \varpi^i(k) \\ \overline{r}(k) \end{bmatrix}^T \Pi_1 \begin{bmatrix} x^i(k) \\ \varpi^i(k) \\ \overline{r}(k) \end{bmatrix} + \min_{\substack{U = \{u_k, \dots, u_{k+N_u-1}\}}} \left\{ 2 \begin{bmatrix} x^i(k) \\ \varpi^i(k) \\ \overline{r}(k) \end{bmatrix}^T \Pi_2 U^i + \frac{1}{2} U^i^T 2 \Pi_3 U^i \right\}.
\end{aligned} \tag{11}$$

The optimization problem of Equation (6) is equivalent to:

г

$$GU \le W + E \begin{bmatrix} x^{i}(k) \\ \omega^{i}(k) \\ \overline{r}(k), \end{bmatrix}$$
(12)

where

$$G = \begin{bmatrix} I & -I & P_{yu}^{i} & -P_{yu}^{i} \end{bmatrix}$$
$$W = \begin{bmatrix} \begin{bmatrix} I \\ \vdots \\ I \end{bmatrix} u_{\max} & -\begin{bmatrix} I \\ \vdots \\ I \end{bmatrix} u_{\min} \begin{bmatrix} I \\ \vdots \\ I \end{bmatrix} y_{\max} & -\begin{bmatrix} I \\ \vdots \\ I \end{bmatrix} y_{\min} \begin{bmatrix} I \\ i \end{bmatrix} y_{$$

Finally, let

$$Z_1 = \begin{bmatrix} x^i(k) \\ \varpi^i(k) \\ \overline{r}(k) \end{bmatrix}$$
$$Z = U^i + (\Pi_3)^{-1} * \Pi_2 Z_1$$
$$S = E + G * (2\Pi_3)^{-1} (2\Pi_2)^T$$

then Equation (6) can now be transformed into a standard mpQP-eMPC:

$$\min_{Z} \frac{1}{2} Z^{T} 2 \Pi_{3} Z$$

$$s.t.GZ \le W + SZ_{1}.$$
(13)

**Remark 2.** The transformation is necessary. The benefit from the standard form of mpQP-eMPC, the deducing procedure of explicit expression in solving the optimization problem based on KKT conditions becomes simpler, more concise and easier to understand.

#### 3.3. Offline Solution Algorithm of Turboshaft Engine Controller Based on mpQP-eMPC

The transformation of the turboshaft engine-rotor system tMPC into the standard mpQP-eMPC optimization problem is completed as above, and the turboshaft engine controller offline step can now proceed by referring to the mpQP-eMPC algorithm procedure.

**Lemma 2** ([15]). For any partition  $\{\vartheta_l\}_{l=0}^L = \{\vartheta_0, \vartheta_1, \cdots, \vartheta_L\}$  of the parameter space  $\vartheta$ , suppose that the optimization problem (13) on  $\vartheta_1$  satisfies both the linear independent constraint condition LICQ and the strict complementary relaxation condition, and (13) has an initial feasible solution

 $Z_{10} = \begin{bmatrix} x_0(k) \\ \omega_0(k) \\ \bar{r}_0(k) \end{bmatrix}$  in  $\vartheta$ , then the optimal solution of the optimization problem (13) is as follows:

$$\begin{bmatrix} Z(x) \\ \lambda(x) \end{bmatrix} = -(M_0)^{-1} N_0 (Z_1 - Z_{10}) + \begin{bmatrix} Z_0 \\ \lambda_0, \end{bmatrix}$$
(14)

where 
$$M_0 = \begin{bmatrix} 2\Pi_3 & G_1^T & \cdots & G_1^T \\ -\lambda_1 G_1 & -V_1 & \cdots & -V_q \\ \vdots & \vdots & \ddots & \vdots \\ -\lambda_p G_q & -V_1 & \cdots & -V_q \end{bmatrix}$$
,  $N_0 = \begin{bmatrix} Y & \lambda_1 S_1 & \cdots & \lambda_p S_p \end{bmatrix}$ ,  $V_i = G_i Z_0 - V_i = V_i =$ 

 $W_i - S_i Z_{10}$ ,  $G_i$ ,  $S_i$  and  $W_i$  represent the *i*th row of matrix G, S, and W respectively. Y is a zero matrix of dimension  $s \times n$ ,  $\lambda_0 = \lambda(x_0)$  is a set of nonnegative Lagrange multipliers,  $Z_0 = Z(Z_{10})$  is a set of initial feasible solutions of optimization problem (13).

From Lemma 2, it is not difficult to find that the key to solving the optimization problem is to determine a partition  $\{\vartheta_0, \vartheta_1, \dots, \vartheta_L\}$  of  $\vartheta$  and the feasible solutions  $Z_{10}$  and  $Z_0$  on any partition  $\vartheta_l$ . The partition of  $\vartheta$  can refer to the method in Lemma 3.

**Lemma 3** ([9]). Suppose  $\vartheta \subseteq \mathbb{R}^d$  is a polytope (polyhedron),  $\vartheta_0 \triangleq \{Z_1 \in \vartheta : H_0Z_1 \leq K_0\}$  is a subset of  $\vartheta \subseteq \mathbb{R}^d$ , and  $\vartheta_0 \neq \emptyset$ . Let  $L = \dim(K_1)$ ,  $\vartheta^{rest} \triangleq \bigcup_{l=1}^L \vartheta_l$ ,

$$\vartheta_l = \{Z_1 \in \vartheta : H_l Z_1 \le K_l\}, \ l = 1, \cdots, L.$$
(15)

*then:* 
$$\{\vartheta_0, \vartheta_1, \cdots, \vartheta_L\}$$
 *is a partition of*  $\vartheta$ *, where*  $H_l = \begin{bmatrix} -T_l \\ T_j \end{bmatrix}$ *,*  $K_l = \begin{bmatrix} -\theta_l \\ \theta_j \end{bmatrix}$ *,*  $\forall j < l$ *.*

Next, the explicit expression of the mpQP-eMPC controller is given through the following theorem.

**Theorem 1.** Assuming that the parameter space partition  $\{\vartheta_1, \vartheta_2, \dots, \vartheta_L\}$  of the perturbed PWA system (2) is obtained based on the method in Lemma 3, where  $F_l'$  is the first *m* rows of matrix  $-(M_0)^{-1}N_0$ ,  $G_l'$  is the first *m* rows of matrix  $(M_0)^{-1}N_0Z_{10} + Z_0$ , and *m* is the number of rows of *Z*, then the mpQP-eMPC controller of the turboshaft engine–rotor system is continuous, and its explicit expression on any partition is as follows:

$$U^{i} = F_{l}Z_{1} + G_{l}, \ F_{l} = F_{l}^{'} - (\Pi_{3})^{-1} * \Pi_{2}, \ G_{l} = G_{l}^{'}.$$
(16)

**Proof.** From Lemma 1, we can see that the mpQP-eMPC controller of the turboshaft engine–rotor system is continuous. Based on Lemma 2,  $Z(Z_1) = F_l'Z_1 + G_l'$ . Because  $Z = U^i + (\Pi_3)^{-1} * \Pi_2 Z_1$ , therefore:

$$U^{i} = \left(F_{l}^{'} - (\Pi_{3})^{-1} * \Pi_{2}\right)Z_{1} + G_{l}^{'} = F_{l}Z_{1} + G_{l}$$

Thus, Theorem 1 is proved.  $\Box$ 

Furthermore, with the help of Chebyshev distance, a feasible solution  $Z_{10}$  of the optimization problem (13) in any partition  $\vartheta_l$  is obtained:

$$\begin{array}{l} \max \\ E_{1,Z,E} \\ S.t. \\ -T_{l}Z_{1} < -\theta_{l} \\ T_{j}Z_{1} \leq \theta_{j}, \forall j < l \\ GZ \leq W + SZ_{1}. \end{array}$$

$$(17)$$

If  $\varepsilon \leq 0$ , then the problem in (13) is infeasible for all  $Z_1$  in  $\vartheta_l$ . Otherwise, we fix  $Z_1 = Z_{10}$  and obtain  $Z_0$  based on the following standard quadratic optimization problem:

 $\begin{array}{l} \min_{Z} \frac{1}{2} Z^{T} H Z \\ s.t. \\ GZ \leq W + SZ_{1} \\ - T_{l} Z_{1} < -\theta_{l} \\ T_{j} Z_{1} \leq \theta_{j}, \forall j < l. \end{array}$ (18)

Finally, based on Lemmas 2 and 3, solutions  $Z_{10}$  and  $Z_0$  of Equations (17) and (18), and Theorem 1, the mpQP-eMPC controller algorithm of the turboshaft engine rotor system is summarized in Algorithm 1.

Algorithm 1 mpQP-eMPC Controller Iteration Algorithm				
Step 1: In the space $GZ \leq W + SZ_1$ , based on the optimization problem (17), find				
the initial value of the parameter $Z_{10}$ ;				
Step 2: Fix $Z_{10}$ , and obtain $\begin{vmatrix} Z_0 \\ \lambda_0 \end{vmatrix}$ by solving quadratic programming				
optimization problem Equation (18);				
Step 3: Determine the critical region $\vartheta_l$ near the equilibrium point $\begin{bmatrix} Z_0 \\ \lambda_0 \end{bmatrix}$ by				
judging whether $\lambda_0$ is greater than 0;				
Step 4: Based on Equation (16) in Theorem 1. to obtain the expressions of				
$\begin{bmatrix} Z(x) \\ \lambda(x) \end{bmatrix}$ and $u(x)$ in the critical region $\vartheta_l$ ;				
Step 5: Divide the critical region $\vartheta_l$ based on Lemma 3 to obtain the subcritical				
region $\vartheta_{li}$ of this critical region;				
Step 6: Repeat the above steps to segment each subcritical domain $\vartheta_{li}$ iteratively,				
and determine the values of $\begin{bmatrix} Z(x) \\ \lambda(x) \end{bmatrix}$ and $u(x)$ in each polyhedron (polyhedron)				
based on Theorem 1.				

### 3.4. Analysis of the Influence of $N_y$ on the Partition Number of Parameter Space

The length of the prediction horizon  $N_y$  has a direct impact on the number of parameter space partitions. In general, the larger the  $N_y$ , the more parameter space partitions, and the longer the time it takes to find the controller during the online query procedure. However, the length of  $N_y$  is not the bigger the better. The case of xcpc = 80 in the turboshaft engine is taken as an example to illustrate this argument.

The linear system at xcpc = 80 (corresponding to the linear system with i = 3 in (1)) is considered:

$$A = \begin{bmatrix} 0.9701 & -2.9772 \times 10^{-4} \\ 0.0011 & 0.9992 \end{bmatrix}, B = \begin{bmatrix} 0.0062 \\ 2.3121 \times 10^{-4} \end{bmatrix}, C = \begin{bmatrix} 1 & 0 \end{bmatrix}, w = \begin{bmatrix} 7.2072 \times 10^{-8} \\ -4.8147 \times 10^{-4} \end{bmatrix}$$
(19)

Note that matrices *A*, *B*, *C* and *w* are the results of normalization and discretizing.

For  $N_y = 3$  and  $N_y = 30$ , respectively, based on the iterative algorithm summarized in Table 1, the explicit model predictive controllers are designed for the system (19) according to the given values in Table 1.

Table 1. Simulation parameter table.

Q	R	T <sub>s</sub>	$\left[\begin{array}{c}n_{g0}\\n_{p0}\end{array}\right]$	$w_{f0}$	xcpc <sub>0</sub>	Constraints
500	0.01	0.01	$\left[\begin{array}{c}42,309\\20,900\end{array}\right]$	0.0747	80	$0.064 \le w_f \le 0.1$ $41,304 \le y \le 43,650$

Among them, Figures 3 and 4 are the simulation results when  $N_y = 3$ ; Figures 5 and 6 show the simulation results when  $N_y = 30$ . It can be seen from Figures 3 and 5 that, when  $N_y = 3$ , the system state space can be divided into five partitions, and when  $N_y = 30$ , the system state space can be divided into 33 partitions. That is to say, the larger the  $N_y$ , the more divisions of the interval division. However, when  $N_y = 30$ , only the number of parameter space partition changes, and the controller value does not change greatly. In this case, the dynamic performance of the controlled plant is consistent with that of  $N_y = 3$ . Therefore, when the turboshaft engine rotor system is considered, the size of  $N_y$  should be selected reasonably according to the dynamic of the system to reduce the online online controller query time.











**Figure 5.** The partition of state space:  $N_{y} = 30$ .



**Figure 6.** Affine function relationship between controller and state partition:  $N_y = 30$ .

## 4. Simulation Example

Under flight condition H = 0, Ma = 0, ten equilibrium points are selected to establish the PWA system for the turboshaft engine–rotor component-level nonlinear model. Some of the PWA models are shown as follows; the others are similar.

Equilibrium point 1 (i = 1): xcpc = 100,  $\begin{bmatrix} n_g & n_p \end{bmatrix} = \begin{bmatrix} 44,679.7259 & 20,900 \end{bmatrix}$ 

$$A^{1} = \begin{bmatrix} 0.9665 & -0.0003\\ 0.0009 & 0.9988 \end{bmatrix} B^{1} = \begin{bmatrix} 0.0061\\ 0.0002 \end{bmatrix} w^{1} = \begin{bmatrix} 1.0258 \times 10^{-7}\\ -0.0006. \end{bmatrix}$$
(20)

Equilibrium point 2 (i = 2): xcpc = 90,  $\begin{bmatrix} n_g & n_p \end{bmatrix} = \begin{bmatrix} 43,393.4576 & 20,900 \end{bmatrix}$ 

$$A^{2} = \begin{bmatrix} 0.9641 & -0.0003\\ 0.0011 & 0.9990 \end{bmatrix} B^{2} = \begin{bmatrix} 0.0066\\ 0.0002 \end{bmatrix} w^{2} = \begin{bmatrix} 8.8320 \times 10^{-8}\\ -0.0006. \end{bmatrix}$$
(21)

Equilibrium point 6 (i = 6): xcpc = 50,  $\begin{bmatrix} n_g & n_p \end{bmatrix} = \begin{bmatrix} 39,059.0789 & 20,900 \end{bmatrix}$ 

$$A^{6} = \begin{bmatrix} 0.9803 & -0.0004\\ 0.0007 & 0.9997 \end{bmatrix} B^{6} = \begin{bmatrix} 0.0069\\ 0.0002 \end{bmatrix} w^{6} = \begin{bmatrix} 2.8358 \times 10^{-8}\\ -0.0002. \end{bmatrix}$$
(22)

Equilibrium point 8 (i = 8): xcpc = 30,  $\begin{bmatrix} n_g & n_p \end{bmatrix} = \begin{bmatrix} 38,007.2583 & 20,900 \end{bmatrix}$ 

$$A^{8} = \begin{bmatrix} 0.9807 & -0.0003\\ 0.0007 & 0.9998 \end{bmatrix} B^{8} = \begin{bmatrix} 0.0075\\ 0.0002 \end{bmatrix} w^{8} = \begin{bmatrix} 2.5422 \times 10^{-8}\\ -0.0001. \end{bmatrix}$$
(23)

Equilibrium point 10 (i = 10): xcpc = 10,  $\begin{bmatrix} n_g & n_p \end{bmatrix} = \begin{bmatrix} 37,119.1035 & 20,900 \end{bmatrix}$ 

$$A^{10} = \begin{bmatrix} 0.9820 & -0.0004\\ 0.0005 & 0.9998 \end{bmatrix} B^{10} = \begin{bmatrix} 0.0078\\ 0.0001 \end{bmatrix} w^{10} = \begin{bmatrix} 1.4611 \times 10^{-8}\\ -6.6602 \times 10^{-5}. \end{bmatrix}$$
(24)

Note that all the matrices appearing in the (20)–(24) are the results of normalization and discretizing.

### 4.1. PWA System

Based on two groups of different input signals, the effectiveness of PWA system modeling is verified.

(1) Figure 7 shows the step input signal of the engine fuel and rotor system (including the large step signal in the first 250s and the small step signal in the last 950s). The speed response curves of PWA system and turboshaft engine–rotor system are shown in Figures 8 and 9. Among them, the maximum overshoot of power turbine speed  $n_p$ 



is 0.9569%, the steady-state modeling error is less than 0.024%. The transition and the steady-state modeling error of gas turbine rotaion speed  $n_g$  are both less than 0.01%.





Figure 8. Speed response curves of power turbine.



Figure 9. Speed response curves of gas turbine.

(2) Figure 10 shows the ramp input signals of engine fuel  $w_f$  and collective pitch *xcpc* of the rotor system. The speed response curves of the PWA system and the turboshaft engine–rotor system are shown in Figures 11 and 12. According to the response curves of power turbine rotation speed  $n_p$  and gas turbine rotation speed  $n_g$ , the steady-state and dynamic errors of  $n_p$  are less than 0.24%; The steady and dynamic error of  $n_g$  is less than 0.01%. Therefore, the output of the PWA system is basically consistent with the

nonlinear model of the turboshaft engine rotor system, which meets the requirements of the controller design.

## 4.2. mPQP-eMPC Controller

In the design process of the control system, by making  $u(k) = u(k-1) + \Delta u(k)$ , an integral unit is added to ensure the steady-state servo tracking without static error, and the fuel change rate limit is introduced into the optimization problem (13) to take into consideration actuator position saturation and rate saturation.



**Figure 10.** Ramp input signals of fuel *w*<sub>f</sub> and collective pitch *xcpc*.



Figure 11. Speed response curves of power turbine.



Figure 12. Speed response curves of gas turbine.

Based on the constraints and parameter settings in Table 2, the explicit model predictive controller is designed according to procedures proposed in the above contents and is compared with the tMPC method, where Q is  $[1,1,1,1,1,1,2,1,0.1,1]^T$  and R is  $[0.1,0.2,0.1,0.2,0.2,0.2,0.5,0.5,0.01,0.1]^T$ .

Table 2. Simulation parameter table.

Ny	$N_y$	$T_s$	$\left[\begin{array}{c}n_{g0}\\n_{p0}\end{array}\right]$	$w_{f0}$	xcpc <sub>0</sub>	Constraints
3	3	0.01	37,119           20,900	0.0319	10	$0.002 \leqslant w_f \leqslant 0.11  30,000 \leqslant y \leqslant 45,000$

The input signal of collective pitch is designed to simulate vertical take off and landing procedure of the helicopeter (Figure 13): During the first 10 s, the collective pitch lever is pushed up quickly to make a rapid step change of *xcpc* value from 10% to 100%. The turboshaft engine will have to transfer from idle working state to maximum power state quickly. This is to simulate the vertical take-off process. After working on *xcpc* = 100% for approximately 100 s, the collective pitch lever is pulled down in two steps to make *xcpx* drop from 100% to 10% which realizes rapid vertical descent process. The helicopter vertical climbing process is also simulated: the collective pitch slowly increases from 10% position to 100% position step by step. The simulation results are shown in Figures 14–17.

Figure 14 shows the fuel flow response curves in two comparative algorithms. The controller based on the mpQP-eMPC algorithm proposed in this paper can ensure that the fuel flow reaches the limit line quickly without exceeding the limit. The fuel rate response curve is shown in Figure 15: the mpQP-eMPC based algorithm can ensure that the fuel rate is close to the limit line as soon as possible but keeps within the limited change rate range in the same time. The system dynamic response performance is thus improved. Figure 16 shows the speed response curve of gas turbine under two comparative algorithms: the mpQP-eMPC based method does not exceed the limited range for the rotation speed of gas turbine, while under the control of tMPC, the rotation speed of the gas turbine violates the speed constraint for a certain period of time. Figure 17 shows the rotation speed response curve of the power turbine under two comparative controllers: the power turbine rotation speed response under the mpQP-eMPC based controller is faster than that under the control of tMPC. The rotation speed overshoot under the mpQP-eMPC controller is also smaller whose maximum value is less than 1.44% compared to 2.4% under the control of tMPC. The steady-state control error is less than 0.005% with the controller proposed in this paper.

### 4.3. Real-Time Verification of Mpqp-Empc

In order to verify the real-time performance of the mpQP-eMPC control algorithm, based on a 3.4 GHz Intel processor, simulation verification is carried out considering the different length prediction time domain. The selection of  $N_y$  is shown in Table 3. In addition, the sampling time  $T_s$  is 0.01 (sec) for simulation setting, and the change of collective pitch *xcpc* is the same as shown above in Figure 13.



Figure 13. The input signal of collective pitch.



**Figure 14.** The response curves of fuel  $w_f$ .



**Figure 15.** The response curves of fuel rate  $\Delta u(k)$ .





Figure 16. Speed response curves of power turbine.



Figure 17. Speed response curves of gas turbine.

**Table 3.** Real time verification table.

$N_y$	Partition Number	Average Con	puting Time	Maximum Computing Time	
	r artition r uniber	mpQP-eMPC	tMPC	mpQP-eMPC	tMPC
3	225	$7.8270 \times 10^{-6}$	$0.4209  imes 10^{-2}$	$3.5050 imes10^{-4}$	$3.9680  imes 10^{-2}$
5	1557	$7.9000  imes 10^{-6}$	$0.4782  imes 10^{-2}$	$2.6660  imes 10^{-4}$	$1.1966  imes 10^{-2}$
7	5892	$9.1960  imes 10^{-6}$	$0.5331  imes 10^{-2}$	$7.1950 imes10^{-4}$	$3.2030  imes 10^{-2}$
9	18,888	$8.0130\times10^{-6}$	$0.5265  imes 10^{-2}$	$2.1990\times 10^{-4}$	$2.7146\times10^{-2}$

The comparison simulation results of the two methods are shown in Table 3, and the comparison simulation results are shown in Figures 18–21.

Figure 18 shows the calculation time of different algorithms at each sampling time when  $N_y = 3$ . The average calculation time of the mpQP-eMPC controller is  $7.8270 \times 10^{-6}$  s, and the maximum calculation time is  $3.5050 \times 10^{-4}$  s; The average computing time of the traditional MPC controller is  $0.4209 \times 10^{-2}$  s, and the maximum computing time is  $3.9680 \times 10^{-2}$  s.

Figure 19 shows the calculation time of different algorithms at each sampling time when  $N_y = 5$ . The average calculation time of the mpQP-eMPC controller is  $7.9000 \times 10^{-6}$  s, and the maximum calculation time is  $2.6660 \times 10^{-4}$  s; the average computing time of the traditional MPC controller is  $0.4782 \times 10^{-2}$  s, and the maximum computing time is  $1.1966 \times 10^{-2}$  s.



**Figure 18.** When  $n_y = 3$ , at each sampling time: (a) mpQP-eMPC calculation time; (b) tMPC calculation time.



**Figure 19.** When  $n_y = 5$ , at each sampling time: (a) mpQP-eMPC calculation time; (b) tMPC calculation time.

Figure 20 shows the calculation time of different algorithms at each sampling time when  $N_y = 7$ . The average calculation time of the mpQP-eMPC controller is  $9.1960 \times 10^{-6}$  s, and the maximum calculation time is  $7.1950 \times 10^{-4}$  s; the average computing time of the traditional MPC controller is  $0.5331 \times 10^{-2}$  s, and the maximum computing time is  $3.2030 \times 10^{-2}$  s.

Figure 21 shows the calculation time of different algorithms in each sampling time when  $N_y = 9$ . The average calculation time of mpQP-eMPC controller is  $8.0130 \times 10^{-6}$  s, and the maximum calculation time is  $2.1990 \times 10^{-4}$  s; the average computing time of the traditional MPC controller is  $0.5265 \times 10^{-2}$  s, and the longest computing time is  $2.7146 \times 10^{-2}$  s.

**Remark 3.** References [16,17] also offer different methods to improve the real-time property, such as Mehdi Hosseinzadeh using the explicit reference governor to enhance the real time character in [16]. Meanwhile, reference [17] has also done a lot of related work on this, and the method proposed in [17] can be seen as an extension of the method in [16]. In addition, there are some robust methods that can improve the real-time performance of the model predictive algorithm [4]. However, considering that inequality scaling is involved and the special research objects, such as turboshaft engines, have strict requirements on the performance of the control system, we have not tried these methods to improve the real-time performance. However, these methods are indeed worth trying, and this may be an important part of our work in the future.



**Figure 20.** When  $n_y = 7$ , at each sampling time: (a) mpQP-eMPC calculation time; (b) tMPC calculation time.



**Figure 21.** When  $n_y = 9$ , at each sampling time: (a) mpQP-eMPC calculation time; (b) tMPC calculation time.

#### 5. Conclusions

A real-time explicit model prediction control for turboshaft engines based on multiparameter quadratic programming is proposed. Firstly, the perturbed PWA system is constructed. Then, based on the theory of PWA and multi-parameter quadratic programming, the mpQP-eMPC controller is designed for turboshaft engine–rotor systems. Finally, the effectiveness of the proposed method is verified by simulating the steady-state and transient process of a turboshaft engine–rotor system with a limit protection requirement. The simulation result proves that: (1) the perturbed PWA linear modeling method is accurate enough; (2) under the proposed mpQP-eMPC based controller, the system dynamic performance is improved while obeying certain constraints at the same time; (3) it breaks through the bottleneck of tMPC finding it difficult to guarantee the real-time performance of turboshaft engines.

There are many ways to improve the real-time performance, but we just tried one of these methods on the turboshaft engine. The focus of future research can be to try a variety of methods to enhance the real-time performance of the MPC algorithm and to finally determine a method that can best improve the control performance of the turboshaft engine.

**Author Contributions:** Conceptualization, N.G.; methodology, N.G.; software, N.G.; validation, N.G.; formal analysis, N.G.; investigation, N.G.; resources, N.G.; data curation, N.G.; writing—original draft preparation, N.G.; writing—review and editing, X.W.; visualization, X.W.; supervision, X.W., N.G. and M.Z.; project administration, X.W.; funding acquisition, X.W. and X.W. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by National Science and Technology Major Project under Grant 2017-V-0015-0067.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

**Data Availability Statement:** The data presented in this study are available on request from the corresponding author.

**Acknowledgments:** The authors would like to thank the support from Collaborative Innovation Center for Advanced Aero-Engine.

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

#### **Parameter Symbol**

ng	Gas turbine rotation speed
$n_{g0}$	Gas turbine rotation speed at equilibrium point
$n_p$	Power turbine rotation speed
$n_{p0}$	Power turbine rotation speed at equilibrium point
хсрс	Collective pitch input of rotor system
xcpc <sub>0</sub>	Collective pitch input at equilibrium point
$w_f$	Engine fuel input
wf0	Engine fuel input at equilibrium point
Ĥ	Helicopter flight altitude
$N_y$	Prediction horizon
Ma	Helicopter flight Mach number
N <sub>u</sub>	Control horizon
MPC	Model predictive control
tMPC	Traditional model predictive control
mpQP-eMPC	Multi-parameter quadratic programming explicit model predictive control
PWA	Piecewise affine system

#### References

- 1. Ballin, M.G. *A High Fidelity Real-Time Simulation of a Small Turboshaft Engine;* NASA Technical Memorandum; National Aeronautics and Space Administration: Washington, DC, USA, 1988.
- 2. Rasmussen, H. Advanced Control of Turbofan Engines; Springer: New York, NY, USA, 2012.
- Mhaskar, P. Robust Model Predictive Control Design for Fault-Tolerant Control of Process Systems. Ind. Eng. Chem. Res. 2006, 45, 8565–8574. [CrossRef]
- 4. Xia, Y.; Liu, G.P.; Shi, P.; Chen, J.; Rees, D. Robust constrained model predictive control based on parameter-dependent Lyapunov functions. *Circuits Syst. Signal Process.* **2008**, 27, 429–446. [CrossRef]
- Wang, T.; Gao, H.; Qiu, J. A Combined Adaptive Neural Network and Nonlinear Model Predictive Control for Multirate Networked Industrial Process Control. *IEEE Trans. Neural Netw. Learn. Syst.* 2017, 27, 416–425. [CrossRef] [PubMed]
- Luo, X.; Yang, Y.U.; Jun, X.U.; Automation, D.O. Online optimization implementation on model predictive control in chemical process. *Ciesc J.* 2014, 65, 3984–3992.
- 7. Tyagunov, A.A. High-Performance Model Predictive Control for Process Industry. Ph.D. Thesis, Technische Universiteit Eindhoven: Eindhoven, The Netherlands, 2004.
- 8. Yang, Y.; Ding, B. Two layer model predictive control for chemical process model with integrating controlled variables. *Can. J. Chem. Eng.* **2019**, *98*, 237–253. [CrossRef]
- 9. Bemporad, A.; Morari, M.; Dua, V.; Pistikopoulos, E.N. The explicit solution of model predictive control via multiparametric quadratic programming. In Proceedings of the 2000 American Control Conference, Chicago, IL, USA, 28–30 June 2000.
- 10. Acevedo, J.; Pistikopoulos, E.N. A Multiparametric Programming Approach for Linear Process Engineering Problems under Uncertainty. *Ind. Eng. Chem. Res.* **1997**, *36*, 717–728. [CrossRef]
- 11. Verlag, S. Advances in Sensitivity Analysis and Parametric Programming; Kluwer Academic Publishers: Amsterdam, The Netherlands, 1998.
- 12. Hovl, S.; Gravdahl, J.T.; Willcox, K.E. Explicit Model Predictive Control for Large-Scale Systems via Model Reduction. *J. Guid. Control Dyn.* **2008**, *31*, 918–926.
- 13. Mariethoz, S.; Domahidi, A.; Morari, M. High-Bandwidth Explicit Model Predictive Control of Electrical Drives. *IEEE Trans. Ind. Appl.* **2012**, *48*, 1980–1992 [CrossRef]
- 14. Wang, Z.J.; Wang, J.Z.; Ma, L.L.; Zhao, J.B. Explicit model predictive control for large inertia servo system. *Trans. Beijing Inst. Technol.* **2011**, *31*, 1307–1312.

- 15. Pistikopoulos, E.N.; Dua, V.; Bozinis, N.A.; Bemporad, A.; Morari, M. On-line optimization via off-line parametric optimization tools. *Comput. Chem. Eng.* 2002, 24, 183–188. [CrossRef]
- 16. Hosseinzadeh, M.; Garone, E. An Explicit Reference Governor for the Intersection of Concave Constraints. *IEEE Trans. Autom. Control* **2019**, *65*, 1–11. [CrossRef]
- 17. Nicotra, M.M.; Liao-Mcpherson, D.; Kolmanovsky, I.V. Dynamically Embedded Model Predictive Control. In Proceedings of the 2018 Annual American Control Conference (ACC), Milwaukee, WI, USA, 27–29 June 2018.