processes

MDPI

Article

Analytical Tuning Method of MPC Controllers for MIMO First-Order Plus Fractional Dead Time Systems

Ning He, Yichun Jiang and Lile He *

School of Mechanical and Electrical Engineering, Xi'an University of Architecture and Technology, Xi'an 710055, China; hening@xauat.edu.cn (N.H.); jycn23@126.com (Y.J.)

* Correspondence: hllnh2013@163.com; Tel.: +86-13474072804

Received: 2 January 2020; Accepted: 6 February 2020; Published: 10 February 2020

Abstract: An analytical model predictive control (MPC) tuning method for multivariable first-order plus fractional dead time systems is presented in this paper. First, the decoupling condition of the closed-loop system is derived, based on which the considered multivariable MPC tuning problem is simplified to a pole placement problem. Given such a simplification, an analytical tuning method guaranteeing the closed-loop stability as well as pre-specified time-domain performance is developed. Finally, simulation examples are provided to show the effectiveness of the proposed method.

Keywords: model predictive control; parameter tuning; analytical method; multivariable fractional dead time system

1. Introduction

Model predictive control (MPC) is widely applied in various kinds of applications in different industrial areas like chemical, petrochemical, automotive, and aerospace [1–3]. As an advanced modern control technology, MPC uses a specific process model to predict future plant response and solves an optimization problem based on such a prediction to obtain the control signal. Such a configuration provides a number of advantages to the MPC system, like robustness, flexibility on model requirements, and easiness for dealing with the constraints on controls and states, which makes the MPC the most successful control method except for PID (Proportion Integration Differentiation) controllers [4,5].

In MPC design, there are several tuning parameters, such as prediction horizon, control horizon, and weight matrices in the cost function. These parameters can significantly influence the closed-loop performance, stability, and robustness of the controlled system. For multivariable systems, parameter tuning becomes more intricate in light of the complex coupling [6]. Therefore, parameter tuning becomes more crucial and challenging for the successful implementation of multivariable MPC. The existing tuning methods in industrial applications are mostly based on practical experiences or numerical methods, which are extraordinarily time-wasting and computationally expensive [7]. In addition, the study of the closed-loop properties cannot be performed conveniently via those methods [8]. Therefore, it is important to study the analytical MPC tuning methodology which gives closed form formulas for the selection of the tuning parameters.

As a large portion of the industrial processes controlled by MPC can be effectively modeled by first-order plus dead time (FOPDT) systems, we focus on MPC tuning for FOPDT systems in this work. In Shridhar and Cooper [9], an analytical tuning strategy is proposed for unconstrained multivariable dynamic matrix control (DMC) through the approximation of the FOPDT model. In Nery Júnior et al. [10], an analytical DMC tuning tool is developed based on the application of analysis of variance (ANOVA) and nonlinear regression analysis for FOPDT systems. In Bagheri and Khakisedigh [11], a constrained multivariable MPC tuning method is developed for uncertain plants using particle swarm optimization technique. In Shah and Engell [12], an approach to determine GPC (Generalized Predictive Control) tuning parameters for unconstrained multiple-input multiple-output (MIMO) system is presented based on the frequency domain analysis using convex optimization solvers. In Yong et al. [13], a generalized predictive control method using the recursive least squares algorithm is presented for the FOPDT model. In Bagheri and Khakisedigh [14], an analytical MPC tuning method is proposed for a single-input single-output (SISO) FOPDT model without active constraints. Moreover, an analytical MPC tuning method for a multiple-input multiple-output (MIMO) FOPDT model is developed in Bagheri and Khakisedigh [15]. In Bagheri and Khakisedigh [16], the authors also provide an MPC tuning method that is able to meet pre-described performance requirements for SISO first-order plus fractional dead time (FOPFDT) models.

Note that some industrial processes own a fractional order dead time and when the dead time is approximated from fractional to integral to apply the existing tuning method, the closed-loop performance can significantly downgrade [17], and thus MPC tuning of first-order plus fractional dead time (FOPFDT) systems should be specifically considered. To the best of the authors' knowledge, MPC tuning of FOPFDT systems has only been studied by [16], who, however, just focused on the SISO case. As MIMO plants play an important role in industrial applications, it is vital to study the MIMO FOPFDT case. Therefore, this paper proposes an analytical MPC tuning approach for MIMO FOPFDT models, and the contributions of the paper are summarized as follows:

(1) The decoupling condition of the MIMO FOPFDT system controlled by MPC is derived, based on which the considered multivariable MPC tuning problem is simplified as a pole placement problem.
(2) Given such a simplified MPC tuning problem, an analytical tuning method is proposed to guarantee the closed-loop stability as well as the pre-specified time-domain performance for the considered system.

This paper is organized in the following way: in Section 2, the problem formulation of MPC tuning for MIMO FOPFDT models is addressed. In Sections 3 and 4, by performing a closed-loop analysis, the decoupled condition is obtained, and then the analytical MPC tuning equations are derived. In Section 5, the effectiveness of the proposed tuning formulas is validated through simulation examples. Finally, in Section 6 the paper closes with some conclusions.

2. Problem Formulation

In this section, the multivariable FOPFDT model is provided and transformed into the state space realization. Then, some tuning bases of the multivariable MPC are presented. To facilitate the analysis, it is first presumed that every manipulated variable (input) influences the control variables (outputs) with the same dead time and dynamic behavior.

Consider a square multivariable plant model as

$$\boldsymbol{y}(s) = \boldsymbol{G}(s)\boldsymbol{u}(s) \tag{1a}$$

where

$$\begin{gathered}\boldsymbol{y}(s) = [y_1(s), y_2(s), \ldots, y_m(s)]^{\mathrm{T}}\ , \boldsymbol{u}(s) = [u_1(s), u_2(s), \ldots, u_m(s)]^{\mathrm{T}}\ , \\ \boldsymbol{G}(s) = \left[g_{ij}(s)\right], \qquad g_{ij}(s) = k_{ij}\frac{e^{-\theta_j s}}{\tau_j s+1} \qquad i, j = 1, 2, \ldots, m\ .\end{gathered} \tag{1b}$$

The fractional dead time model is obtained by discretizing from Equation (1a) with a sampling time T_s

$$\boldsymbol{y}(z) = \boldsymbol{G}(z)\boldsymbol{u}(z), \tag{2a}$$

where

$$\begin{gathered}\boldsymbol{y}(z) = [y_1(z), y_2(z), \ldots, y_m(z)]^{\mathrm{T}}, \quad \boldsymbol{u}(z) = [u_1(z), u_2(z), \ldots, u_m(z)]^{\mathrm{T}}, \\ \boldsymbol{G}(z) = [g_{ij}(z)], \qquad g_{ij}(z) = k_{ij}(1-a_j)\frac{(1-b_j)z+b_j}{z^{d_j+1}(z-a_j)} \qquad i, j = 1, 2, \ldots, m,\end{gathered} \tag{2b}$$

and $y_i(z)$ is the ith plant model output and $u_j(z)$ is the jth control signal. $a_i = e^{-T_s/\tau_i}$, $\theta_i = T_s(d_i + b_i)$, $0 \leq b_i < 1$, and d_i is the delay between the ith output and input which is a non-negative integer number. The time delay is generally a non-integer multiple of the sampling time, so the dead time of the discrete model is non-integral. Let $\boldsymbol{K}$ be the gain matrix of Equation (2b) as

$$\boldsymbol{K} = [k_{ij}] \ , \ i, j = 1, 2, \ldots, m \tag{3}$$

Let $\boldsymbol{P}(z)$ and $\boldsymbol{D}(z)$ be the dynamic response and time delay matrices of Equations (2b) and (3), which are represented as

$$\begin{gathered} \boldsymbol{P}(z) = \mathrm{diag}\Big\{(1-a_1)\tfrac{(1-b_1)z+b_1}{z(z-a_1)}, (1-a_2)\tfrac{(1-b_2)z+b_2}{z(z-a_2)}, \ \ldots, (1-a_m)\tfrac{(1-b_m)z+b_m}{z(z-a_m)}\Big\}, \\ \boldsymbol{D}(z) = \mathrm{diag}\Big\{z^{-d_1}, z^{-d_2}, \ldots, z^{-d_m}\Big\}. \end{gathered} \tag{4}$$

Thus,

$$\boldsymbol{G}(z) = \boldsymbol{K}\boldsymbol{P}(z)\boldsymbol{D}(z). \tag{5}$$

Then, the following state space realization of Equation (2a) is obtained,

$$\begin{gathered} \boldsymbol{x}(n+1) = \boldsymbol{A}\boldsymbol{x}(n) + \boldsymbol{B}\boldsymbol{u}(n) \\ \boldsymbol{y}(n) = \boldsymbol{C}\boldsymbol{x}(n) \end{gathered} \tag{6a}$$

where

$$\boldsymbol{y}(n) = [y_1(n), y_2(n), \ldots, y_m(n)]^{\mathrm{T}} \quad , \boldsymbol{u}(n) = [u_1(n), u_2(n), \ldots, u_m(n)]^{\mathrm{T}}, \tag{6b}$$

and depending on whether dead times of subsystems are the same or not, the derivations of the vectors and matrices $\boldsymbol{x}(n)$, $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{C}$ are not the same, and therefore the details are represented later in Sections 3 and 4 separately when the tuning problem for each case is studied.

Define the prediction of model outputs as

$$\hat{\boldsymbol{y}}(n) = \boldsymbol{T}\boldsymbol{x}(n) + \boldsymbol{L}\overline{\boldsymbol{u}}(n), \tag{7}$$

where $\hat{\boldsymbol{y}}(n)$ is the prediction of model output, $\overline{\boldsymbol{u}}(n)$ is the corresponding control signal, and the matrices $\boldsymbol{T}$ and $\boldsymbol{L}$ are defined later in Sections 3 and 4. In addition,

$$\begin{aligned} &\hat{\boldsymbol{y}}(n) = [\hat{\boldsymbol{y}}_1{}^{\mathrm{T}}(n), \hat{\boldsymbol{y}}_2{}^{\mathrm{T}}(n), \ldots, \hat{\boldsymbol{y}}_m{}^{\mathrm{T}}(n)]^{\mathrm{T}}, \ \ \hat{\boldsymbol{y}}_i(n) = \ [\hat{y}_i(n+d_i+1|n), \hat{y}_i(n+d_i+2|n), \ldots, \hat{y}_i(n+d_i+P_i|n)]^{\mathrm{T}}, \\ &\overline{\boldsymbol{u}}(n) = [\boldsymbol{u}_1{}^{\mathrm{T}}(n), \boldsymbol{u}_2{}^{\mathrm{T}}(n), \ldots, \boldsymbol{u}_m{}^{\mathrm{T}}(n)], \quad \ \ \boldsymbol{u}_i(n) = [u_i(n), u_i(n+1), \ldots, u_i(n+M_i-1)]^{\mathrm{T}} \quad \ \ i = 1, 2, \ldots, m, \end{aligned} \tag{8}$$

where $\hat{\boldsymbol{y}}_i(.|n)$ is the prediction of the ith model output at instance n, P_i is the prediction horizon of the ith output, and M_i is the ith control horizon.

Meanwhile, the prediction of plant outputs are

$$\begin{gathered} \hat{\boldsymbol{y}}_p(n) = [\hat{\boldsymbol{y}}^{\mathrm{T}}{}_{p1}(n), \hat{\boldsymbol{y}}^{\mathrm{T}}{}_{p2}(n), \ldots, \hat{\boldsymbol{y}}^{\mathrm{T}}{}_{pm}(n)]^{\mathrm{T}} = \hat{\boldsymbol{y}}(n) + (\mathrm{diag}\Big\{\boldsymbol{1}_{P_1\times 1}, \boldsymbol{1}_{P_2\times 1}, \ldots, \boldsymbol{1}_{P_m\times 1}\Big\})\boldsymbol{b}(n) \quad , \quad \boldsymbol{1} = [1, 1, \ldots, 1]^{\mathrm{T}}, \\ \hat{\boldsymbol{y}}_{pi}(n) = \ [\hat{y}_{pi}(n+d_i+1|n), \hat{y}_{pi}(n+d_i+2|n), \ldots, \hat{y}_{pi}(n+d_i+P_i|n)]^{T} \quad , \quad \quad i = 1, 2, \ldots, m, \end{gathered} \tag{9}$$

where $\hat{y}_{pi}(\cdot|n)$ is the prediction of the ith plant output at instance n and $b(n)$ is the bias term as follows

$$\boldsymbol{b}(n) = [b_1(n), b_2(n), \ldots, b_m(n)]^{\mathrm{T}} = \boldsymbol{y}_p(n) - \boldsymbol{y}(n) \qquad , \boldsymbol{y}_p(n) = [y_{p1}(n), y_{p2}(n), \ldots, y_{pm}(n)]^{\mathrm{T}}, \tag{10}$$

where $y_{pi}(n)$ is the ith plant output.

Based on the aforementioned descriptions, the optimization problem of predictive control is described as follows

$$
\begin{array}{l}
\min\limits_{\overline{\boldsymbol{u}}(n)}(\overline{\boldsymbol{w}}(n)-\hat{\boldsymbol{y}}_p(n))^{\mathrm{T}}\boldsymbol{Q}(\overline{\boldsymbol{w}}(n)-\hat{\boldsymbol{y}}_p(n))+(\boldsymbol{u}_{ss}-\overline{\boldsymbol{u}}(n))^{\mathrm{T}}\boldsymbol{R}(\boldsymbol{u}_{ss}-\overline{\boldsymbol{u}}(n)) \\
s.t. \quad u_{i_{\min}} \le u_i(n+j) \le u_{i_{\max}}, \quad j=0,1,\ldots,M_i-1 \quad , \quad i=1,2,\ldots,m \\
y_{i_{\min}} \le \hat{y}_{pi}(n+j|n) \le y_{i_{\max}}, \quad j=d_i+1,d_i+2,\ldots,d_i+P_i \quad , \quad i=1,2,\ldots,m
\end{array} \tag{11a}
$$

where

$$
\begin{array}{c}
\overline{\boldsymbol{w}}(n)=(\mathrm{diag}\{\mathbf{1}_{P_1\times1},\mathbf{1}_{P_2\times1},\ldots,\mathbf{1}_{P_m\times1}\})\boldsymbol{w}(n), \quad \boldsymbol{w}(n)=[w_1(n),w_2(n),\ldots,w_m(n)]^T, \\
\boldsymbol{u}_{ss}(n)=(\mathrm{diag}\{\mathbf{1}_{M_1\times1},\mathbf{1}_{M_2\times1},\ldots,\mathbf{1}_{M_m\times1}\})\boldsymbol{K}^{-1}(\boldsymbol{w}(n)-\boldsymbol{b}(n)).
\end{array} \tag{11b}
$$

The matrix $\boldsymbol{Q}$ is positive semi-definite, the matrix $\boldsymbol{R}$ is positive definite, $w_i(.)$ is the reference signal of the ith output and $\boldsymbol{u}_{ss}(n)$ is the steady state value of the control signals.

On the condition of no active constraints, the optimal control solution of Equation (11a) is represented as

$$
\begin{array}{rl}
\overline{\boldsymbol{u}}(n)= & (\boldsymbol{R}+\boldsymbol{L}^{\mathrm{T}}\boldsymbol{QL})^{-1}(\boldsymbol{L}^{\mathrm{T}}\boldsymbol{Q}(\mathrm{diag}\{\mathbf{1}_{P_1\times1},\mathbf{1}_{P_2\times1},\ldots,\mathbf{1}_{P_m\times1}\})[\boldsymbol{w}(n)-\boldsymbol{b}(n)]-\boldsymbol{L}^{\mathrm{T}}\boldsymbol{Q}\overline{\boldsymbol{T}}_y(\overline{\boldsymbol{x}}_y(n)+\boldsymbol{H}\overline{\boldsymbol{x}}_u(n)) \\
& +\boldsymbol{RK}^{-1}(\mathrm{diag}\{\mathbf{1}_{M_1\times1},\mathbf{1}_{M_2\times1},\ldots,\mathbf{1}_{M_m\times1}\})[\boldsymbol{w}(n)-\boldsymbol{b}(n)]),
\end{array} \tag{12}
$$

where $\overline{\boldsymbol{T}}_y$ and $\boldsymbol{H}$ are derived from $\boldsymbol{T}$, and $\overline{\boldsymbol{x}}_y(n)$ and $\overline{\boldsymbol{x}}_u(n)$ present the corresponding parts of the state $\boldsymbol{x}(n)$. The detailed derivation of the above formula is given in Sections 3 and 4. Let

$$
\begin{array}{c}
\overline{\boldsymbol{\Omega}}=(\boldsymbol{R}+\boldsymbol{L}^{\mathrm{T}}\boldsymbol{QL})^{-1}\boldsymbol{L}^{\mathrm{T}}\boldsymbol{Q}\overline{\boldsymbol{T}}_y, \\
\overline{\boldsymbol{\Gamma}}=(\boldsymbol{R}+\boldsymbol{L}^{\mathrm{T}}\boldsymbol{QL})^{-1}(\boldsymbol{L}^{\mathrm{T}}\boldsymbol{Q}(\mathrm{diag}\{\mathbf{1}_{P_1\times1},\mathbf{1}_{P_2\times1},\ldots,\mathbf{1}_{P_m\times1}\})+\boldsymbol{R}(\mathrm{diag}\{\mathbf{1}_{M_1\times1},\mathbf{1}_{M_2\times1},\ldots,\mathbf{1}_{M_m\times1}\}\boldsymbol{K}^{-1}).
\end{array} \tag{13}
$$

Hence, Equation (12) turns to

$$
\overline{\boldsymbol{u}}(n)=\overline{\boldsymbol{\Gamma}}(\boldsymbol{w}(n)-\boldsymbol{b}(n))-\overline{\boldsymbol{\Omega}}(\overline{\boldsymbol{x}}_y(n)+\boldsymbol{H}\overline{\boldsymbol{x}}_u(n)). \tag{14}
$$

Then, the actual optimal control signals of the plant are obtained as

$$
\boldsymbol{u}(n)=\Gamma(\boldsymbol{w}(n)-\boldsymbol{b}(n))-\boldsymbol{\Omega}(\overline{\boldsymbol{x}}_y(n)+\boldsymbol{H}\overline{\boldsymbol{x}}_u(n)) \tag{15a}
$$

where

$$
\begin{array}{c}
\boldsymbol{\Gamma}=\boldsymbol{J}\overline{\boldsymbol{\Gamma}} \quad ,\boldsymbol{\Omega}=\boldsymbol{J}\overline{\boldsymbol{\Omega}}, \\
\boldsymbol{J}=[\boldsymbol{J}_1,\boldsymbol{J}_2,\ldots,\boldsymbol{J}_m] \quad ,\boldsymbol{J}_i=\begin{bmatrix} & & \mathbf{0}_{(i-1)\times M_i} & & \\ 1 & 0 & 0 & \cdots & 0 \\ & & \mathbf{0}_{(m-1)\times M_i} & & \end{bmatrix}_{m\times M_i} \quad i=1,2,\ldots,m.
\end{array} \tag{15b}
$$

Note that Equation (15a) indicates that the control signal is determined by the gain matrices $\boldsymbol{\Omega}$ and $\boldsymbol{\Gamma}$, and therefore via selecting them to desired values (named as $\boldsymbol{\Omega}_d$ and $\boldsymbol{\Gamma}_d$ hereafter), we can acquire the desired control signal. Therefore, MPC tuning problem is solved by analyzing the correspondence between the gain matrices and the tuning parameters ($\boldsymbol{Q}$, $\boldsymbol{R}$, P_i, and M_j). Additionally, it can be shown that $\boldsymbol{\Gamma}$ is relative to $\boldsymbol{\Omega}$ and thus $\boldsymbol{\Omega}$ is chosen as the tuning parameter. However, it is worthwhile to mention that under such a structure, not all desired performances of the multivariable FOPFDT system are achievable, and moreover, some gain matrices that satisfy the achievable performance may not ensure the closed-loop stability.

Based on the above analysis, the tuning problem is described as follows: selecting the weighting matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$, and prediction and control horizons P_i and M_j to achieve the desired gain matrix $\boldsymbol{\Omega}_d$ such that the closed-loop stability and desired performance are guaranteed. To better solve such a problem, FOPFDT systems with same dead time are first considered, and then the obtained results are generalized to the different dead time case.

3. Tuning Method for MIMO FOPDT Models with Same Fractional Dead Time

In this section, the optimal solution to the MPC is derived and represented in the state space realization for FOPFDT systems with the same fractional dead time ($d_1 = d_2 = \cdots = d_m \;\;,\;\; b_1 = b_2 = \cdots = b_m$). Then, the closed-loop transfer function is provided, and the decoupling condition is derived. Finally, the tuning equations are obtained for the desired performance.

3.1. MPC Solution

The ith output of the considered system is represented as

$$y_i(n) = \sum_{j=1}^{m} k_{ij} v_j(n) \qquad\qquad i = 1,2,\ldots,m, \tag{16a}$$

where

$$\begin{array}{c} v_i(n) = (1-a_i)\frac{(1-b_i)z+b_i}{z(z-a_i)} \qquad\qquad i = 1,2,\ldots,m, \\ b_1 = b_2 = \cdots = b_m. \end{array} \tag{16b}$$

The relationship between each $v_i(n)$ for $i = 1, 2, \ldots, m$ and any input is described as

$$\begin{array}{c} \boldsymbol{x}_i(n+1) = \overline{\boldsymbol{A}}_i \boldsymbol{x}_i(n) + \overline{\boldsymbol{B}}_i \boldsymbol{u}(n) \\ \boldsymbol{v}_i(n) = \overline{\boldsymbol{C}}_i \boldsymbol{x}_i(n) \end{array} \tag{17a}$$

where

$$\begin{array}{c} \boldsymbol{x}_i(n) = [v_i(n)\;, v_i(n+1), \cdots, v_i(n+d_i)\;, (1-a_i)b_i u_i(n-1)], \\ \overline{\boldsymbol{A}}_i = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 1 & 0 \\ 0 & 0 & \cdots & 0 & a_i & 1 \\ 0 & 0 & \cdots & 0 & 0 & 0 \end{bmatrix}, \overline{\boldsymbol{B}}_i = (1-a_i) \begin{array}{c} \begin{bmatrix} 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 1-b_i & 0 & \cdots & 0 \\ 0 & \cdots & 0 & b_i & 0 & \cdots & 0 \end{bmatrix} \\ \begin{array}{ccccccc} & & & i\mathrm{Th} & & & \end{array} \end{array}, \\ \overline{\boldsymbol{C}}_i = [\; 1 \quad 0 \quad 0 \quad \cdots \quad 0 \quad 0 \;]. \end{array} \tag{17b}$$

Therefore,

$$\begin{array}{c} \boldsymbol{x}(n) = \left[\boldsymbol{x}_1(n)^{\mathrm{T}}, \boldsymbol{x}_2(n)^{\mathrm{T}}, \ldots, \boldsymbol{x}_m(n)^{\mathrm{T}}\right]^{\mathrm{T}}, \quad \boldsymbol{A} = \mathrm{diag}\left\{\overline{\boldsymbol{A}}_1, \overline{\boldsymbol{A}}_2, \ldots, \overline{\boldsymbol{A}}_m\right\}, \\ \boldsymbol{B} = \left[\overline{\boldsymbol{B}}_1{}^{\mathrm{T}}, \overline{\boldsymbol{B}}_2{}^{\mathrm{T}}, \ldots, \overline{\boldsymbol{B}}_m{}^{\mathrm{T}}\right]^{\mathrm{T}}, \quad \boldsymbol{C} = \boldsymbol{K}\mathrm{diag}\left\{\; \overline{\boldsymbol{C}}_1, \; \overline{\boldsymbol{C}}_2, \ldots, \; \overline{\boldsymbol{C}}_m\right\}. \end{array} \tag{18}$$

The above state space realization is controllable and observable. From Equation (7), the prediction of the future model output value is obtained as

$$\begin{array}{c} \boldsymbol{T} = [\boldsymbol{T}_{ij}]\;, \\ \boldsymbol{T}_{ij} = k_{ij} \begin{bmatrix} \overline{\boldsymbol{C}}_j \overline{\boldsymbol{A}}_j{}^{d_i+1} \\ \overline{\boldsymbol{C}}_j \overline{\boldsymbol{A}}_j{}^{d_i+2} \\ \vdots \\ \overline{\boldsymbol{C}}_j \overline{\boldsymbol{A}}_j{}^{d_i+P_i} \end{bmatrix} = [\boldsymbol{0}, \boldsymbol{0}, \ldots, \boldsymbol{0}, \overline{\boldsymbol{T}}_{y_{ij}}, \overline{\boldsymbol{T}}_{u_{ij}}]\;, \quad i,j = 1,2,\ldots,m, \end{array} \tag{19a}$$

where

$$\overline{\boldsymbol{T}}_{y_{ij}} = a_j \overline{\boldsymbol{T}}_{u_{ij}} = k_{ij}[a_j, a_j{}^2, \cdots, a_j{}^{P_i}]^{\mathrm{T}} \quad i, j = 1, 2, \ldots, m\,, \tag{19b}$$

and **0** is a zero matrix with corresponding dimensions.

$$\boldsymbol{L} = [L_{ij}] \quad , \quad i, j = 1, 2, \ldots, m, \tag{20a}$$

$$L_{ij} = k_{ij}(1-a_j)\left\{(1-b_j)\begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ a_j & 1 & \ddots & \vdots & \vdots \\ \vdots & a_j & \ddots & 0 & \vdots \\ \vdots & \vdots & \ddots & 1 & 0 \\ \vdots & \vdots & \cdots & a_j & 1 \\ \vdots & \vdots & \cdots & \vdots & 1+a_j \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ a_j{}^{P_i-1} & a_j{}^{P_i-2} & \cdots & a_j{}^{P_i-M_j+1} & 1+a_j+\cdots+a_j{}^{P_i-M_j} \end{bmatrix} + b_i \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \\ a_j & 1 & \ddots & \vdots & \vdots \\ \vdots & a_j & \ddots & 0 & \vdots \\ \vdots & \vdots & \ddots & 1 & 0 \\ \vdots & \vdots & \cdots & a_j & 1 \\ \vdots & \vdots & \cdots & \vdots & 1+a_j \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ a_j{}^{P_i-2} & a_j{}^{P_i-3} & \cdots & a_j{}^{P_i-M_j} & 1+a_j+a_j{}^{P_i-M_j-1} \end{bmatrix}\right\}, \tag{20b}$$

$$i, j = 1, 2, \ldots, m.$$

Let

$$\overline{\boldsymbol{T}}_y = [\overline{\boldsymbol{T}}_{y_{ij}}] \quad , \overline{\boldsymbol{T}}_u = [\overline{\boldsymbol{T}}_{u_{ij}}] \qquad i, j = 1, 2, \ldots, m. \tag{21a}$$

Then, $\overline{\boldsymbol{T}}_y \overline{\boldsymbol{x}}_y(n) + \overline{\boldsymbol{T}}_u \overline{\boldsymbol{x}}_u(n)$ is used to represent the active part of.

It can be shown that

$$\overline{\boldsymbol{T}}_u = \overline{\boldsymbol{T}}_y \boldsymbol{H} = \overline{\boldsymbol{T}}_y \mathrm{diag}\left\{a_1{}^{-1}, a_2{}^{-1}, \ldots, a_m{}^{-1}\right\}. \tag{21b}$$

Therefore, from Equation (21b), we have

$$\overline{\boldsymbol{T}}_y \overline{\boldsymbol{x}}_y(n) + \overline{\boldsymbol{T}}_u \overline{\boldsymbol{x}}_u(n) = \overline{\boldsymbol{T}}_y(\overline{\boldsymbol{x}}_y(n) + \boldsymbol{H}\overline{\boldsymbol{x}}_u(n)). \tag{21c}$$

Using Equations (16a) and (21b), it can be obtained that

$$\begin{gathered} \overline{\boldsymbol{x}}_y(n) = \boldsymbol{D}(z)^{-1}\boldsymbol{K}^{-1}\boldsymbol{y}(n), \\ \boldsymbol{H}\overline{\boldsymbol{x}}_u(n) = \boldsymbol{E}\boldsymbol{u}(n-1) = \boldsymbol{E}Z(\boldsymbol{z})\boldsymbol{u}(n), \end{gathered} \tag{22a}$$

where

$$\begin{gathered} \boldsymbol{E} = \mathrm{diag}\{(1-a_1)(b_1/a_1), (1-a_2)(b_2/a_2), \ldots, (1-a_m)(b_m/a_m)\}\,, \\ \boldsymbol{Z}(z) = \mathrm{diag}\{\underbrace{z^{-1}, z^{-1}, \ldots, z^{-1}}_{m}\}. \end{gathered} \tag{22b}$$

Hence, the optimal solution in Equation (15a) is transformed into

$$\boldsymbol{u}(n) = \boldsymbol{\Gamma}(\boldsymbol{w}(n) - \boldsymbol{b}(n)) - \boldsymbol{\Omega}(\boldsymbol{D}(z)^{-1}\boldsymbol{K}^{-1}\boldsymbol{y}(n) + \boldsymbol{E}\boldsymbol{Z}(z)\boldsymbol{u}(n)), \tag{23}$$

where

$$\boldsymbol{\Gamma} = \left[\gamma_{ij}\right] \ , \boldsymbol{\Omega} = \left[\delta_{ij}\right] \quad , \quad i, j = 1, 2, \ldots, m. \tag{24}$$

3.2. Closed-Loop Analysis

Let $b(n) = 0$, and thus the plant and model outputs are the same. Using (23), we have

$$(\boldsymbol{I} + \boldsymbol{K}\boldsymbol{P}(z)\boldsymbol{D}(z)\boldsymbol{\Omega}\boldsymbol{D}(z)^{-1}\boldsymbol{K}^{-1} + \boldsymbol{\Omega}\boldsymbol{E}\boldsymbol{Z}(z))\boldsymbol{y}_p(n) = \boldsymbol{K}\boldsymbol{P}(z)\boldsymbol{D}(z)\boldsymbol{\Gamma}\boldsymbol{w}(n). \tag{25}$$

It can be found that

$$\boldsymbol{\Gamma} = (\boldsymbol{I} + \boldsymbol{\Omega} + \boldsymbol{\Omega E})\boldsymbol{K}^{-1}. \tag{26}$$

Hence, the closed-loop model can be obtained as

$$\boldsymbol{y}_p(n) = \boldsymbol{G}_{cl}(z)\boldsymbol{w}(n), \tag{27a}$$

where

$$\boldsymbol{G}_{cl}(z) = \boldsymbol{KP}(z)\boldsymbol{D}(z)(\boldsymbol{I} + \boldsymbol{P}(z)\boldsymbol{\Omega} + \boldsymbol{\Omega EZ}(z))^{-1}(\boldsymbol{I} + \boldsymbol{\Omega} + \boldsymbol{\Omega E})\boldsymbol{K}^{-1}. \tag{27b}$$

Based on the aforementioned analysis, the first main result of the paper is provided in the following.

Theorem 1. *On the condition of no active constraints and no model mismatch, the tuning problem can be converted to the following decoupled closed-loop transfer function matrix and then restated as a pole placement problem:*

$$\boldsymbol{G}_{cl_d}(z) = \mathrm{diag}\{\frac{\Delta_1}{z^{-(d_1+1)}(z-x_1)}, \quad \frac{\Delta_2}{z^{-(d_2+1)}(z-x_2)}, \ldots, \frac{\Delta_m}{z^{-(d_m+1)}(z-x_m)}\}, \tag{28a}$$

if the gain matrix is selected as

$$\boldsymbol{\Omega} = (\boldsymbol{K}^{-1}[z^d\boldsymbol{G}_{cl_d}(z)]^{-1}\boldsymbol{K} - \boldsymbol{P}(z)^{-1})[\boldsymbol{I} + \boldsymbol{EZ}(z)\boldsymbol{P}(z)^{-1} - \boldsymbol{K}^{-1}[z^d\boldsymbol{G}_{cl_d}(z)]^{-1}\boldsymbol{K}(\boldsymbol{I}+\boldsymbol{E})]^{-1}, \tag{28b}$$

where

$$\Delta_i = (1-a_i)(1+\omega+\omega(1-a_i)b_i a_i{}^{-1})[(1-b_i)z + b_i] \quad i = 1, 2, \ldots, m. \tag{28c}$$

Proof. Letting the closed-loop transfer function derived in Equation (27b) equal the decoupled transfer function in Equation (28a), and solving such an equation for Ω, Equations (28b) and (28c) are then obtained based on some tedious but straightforward calculations. □

3.3. Tuning Equations for Desired Performance

Assuming every dead time is the same and the control horizon is chosen as one, we derive the parameter tuning formulas that achieve the desired performance through the above pole placement problem. The details are summarized in the following theorem.

Theorem 2. *On the condition of no active constraints and no model mismatch, the achievable gain matrix corresponding to the desired performance for the MPC tuning problem should satisfy the following inequalities:*

$$\boldsymbol{\Omega}_d < (\boldsymbol{L}^{\mathrm{T}}\boldsymbol{QL})^{-1}\boldsymbol{L}^{\mathrm{T}}\boldsymbol{Q}\overline{\boldsymbol{T}}_y, \tag{29}$$

and the closed-loop stability condition by the Jury criterion [18].

Choosing $\boldsymbol{R}$ as the tuning parameter, the tuning formula for satisfying the feasible gain matrix $\boldsymbol{\Omega}_d$ is

$$\boldsymbol{R} = \boldsymbol{L}^{\mathrm{T}}\boldsymbol{Q}[\overline{\boldsymbol{T}}_y\boldsymbol{\Omega}_d{}^{-1} - \boldsymbol{L}]. \tag{30}$$

Proof. According to Equation (27b), the closed-loop model can reach the desired form by properly selecting the gain matrix $\boldsymbol{\Omega}_d$. Using Equations (13) and (24), we have

$$\boldsymbol{\Omega}_d = (\boldsymbol{R} + \boldsymbol{L}^{\mathrm{T}}\boldsymbol{QL})^{-1}\boldsymbol{L}^{\mathrm{T}}\boldsymbol{Q}\overline{\boldsymbol{T}}_y, \tag{31}$$

From Equation (29), we have

$$(\boldsymbol{R}+\boldsymbol{L}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{L})\boldsymbol{\Omega}_d=\boldsymbol{L}^{\mathrm{T}}\boldsymbol{Q}\overline{\boldsymbol{T}}_y \Rightarrow (\boldsymbol{R}+\boldsymbol{L}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{L})=\boldsymbol{L}^{\mathrm{T}}\boldsymbol{Q}\overline{\boldsymbol{T}}_y{\boldsymbol{\Omega}_d}^{-1} \Rightarrow \boldsymbol{R}=\boldsymbol{L}^{\mathrm{T}}\boldsymbol{Q}[\overline{\boldsymbol{T}}_y{\boldsymbol{\Omega}_d}^{-1}-\boldsymbol{L}]. \tag{32}$$

Since $\boldsymbol{R}$ is a positive definite matrix, it is obtained that

$$\boldsymbol{L}^{\mathrm{T}}\boldsymbol{Q}[\overline{\boldsymbol{T}}_y{\boldsymbol{\Omega}_d}^{-1}-\boldsymbol{L}]>0 \quad\Rightarrow\quad \boldsymbol{L}^{\mathrm{T}}\boldsymbol{Q}\overline{\boldsymbol{T}}_y{\boldsymbol{\Omega}_d}^{-1}-\boldsymbol{L}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{L}>0 \quad\Rightarrow \boldsymbol{\Omega}_d<(\boldsymbol{L}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{L})^{-1}\boldsymbol{L}^{\mathrm{T}}\boldsymbol{Q}\overline{\boldsymbol{T}}_y. \tag{33}$$

□

4. Tuning Method for MIMO FOPFDT with Different Fractional Dead Times

In this section, the tuning method proposed in the last section is generalized to the different dead time case for the desired performance.

4.1. MPC Solution

Let $\widetilde{\boldsymbol{G}}(z)$ describe the non-delay part of the transfer function matrix

$$\widetilde{\boldsymbol{G}}(z)=[\widetilde{g}_{ij}(z)],\quad \widetilde{g}_{ij}(z)=k_{ij}(1-a_j)\frac{(1-b_j)z+b_j}{z(z-a_j)}\qquad i,j=1,2,\ldots,m. \tag{34}$$

Recall the ith output is

$$y_i(n)=\sum_{j=1}^{m}v_{ij}(n)\qquad i=1,2,\ldots,m, \tag{35a}$$

where

$$v_{ij}(n)=\widetilde{g}_{ij}(z)u_j(n-d_j)\quad i,j=1,2,\ldots,m. \tag{35b}$$

Then, the corresponding state space realization is described as

$$\begin{array}{c}\boldsymbol{x}_{ij}(n+1)=\widetilde{\boldsymbol{A}}_j\boldsymbol{x}_{ij}(n)+\widetilde{\boldsymbol{B}}_{ij}\boldsymbol{u}(n)\\ \boldsymbol{v}_{ij}(n)=\widetilde{\boldsymbol{C}}_j\boldsymbol{x}_{ij}(n)\end{array} \tag{36a}$$

where

$$\begin{array}{c}\boldsymbol{x}_{ij}(n)=[v_{ij}(n)\;,v_{ij}(n+1),\cdots,v_{ij}(n+d_j)\;,k_{ij}(1-a_j)b_ju_j(n-1)],\\
\widetilde{\boldsymbol{A}}_j=\begin{bmatrix}0&1&0&\cdots&0&0\\0&0&1&\ddots&\vdots&\vdots\\\vdots&\vdots&\ddots&\ddots&0&0\\0&0&\cdots&0&1&0\\0&0&\cdots&0&a_j&1\\0&0&\cdots&0&0&0\end{bmatrix},\widetilde{\boldsymbol{B}}_{ij}=k_{ij}(1-a_j)\begin{array}{c}\begin{bmatrix}0&\cdots&0&0&0&\cdots&0\\0&\cdots&0&0&0&\cdots&0\\\vdots&&\vdots&\vdots&\vdots&&\vdots\\0&\cdots&0&0&0&\cdots&0\\0&\cdots&0&1-b_j&0&\cdots&0\\0&\cdots&0&b_j&0&\cdots&0\end{bmatrix}\\ j\mathrm{T}h\end{array},\\
\widetilde{\boldsymbol{C}}_j=[\;1\quad 0\quad 0\quad\cdots\quad 0\quad 0\;],\quad i,j=1,2,\ldots,m.\end{array} \tag{36b}$$

Therefore,

$$\begin{array}{c}\boldsymbol{x}(n)=[\boldsymbol{x}_{11}(n)^{\mathrm{T}},\ldots,\boldsymbol{x}_{1m}(n)^{\mathrm{T}},\boldsymbol{x}_{21}(n)^{\mathrm{T}},\ldots,\boldsymbol{x}_{2m}(n)^{\mathrm{T}},\ldots,\boldsymbol{x}_{m1}(n)^{\mathrm{T}},\ldots,\boldsymbol{x}_{mm}(n)^{\mathrm{T}}]^{\mathrm{T}},\\
\boldsymbol{A}=\mathrm{diag}\left\{\hat{\boldsymbol{A}}_1,\hat{\boldsymbol{A}}_2,\ldots,\hat{\boldsymbol{A}}_m\right\},\\
\boldsymbol{B}=[\widetilde{\boldsymbol{B}}_{11}{}^{\mathrm{T}},\ldots,\widetilde{\boldsymbol{B}}_{1m}{}^{\mathrm{T}},\widetilde{\boldsymbol{B}}_{21}{}^{\mathrm{T}},\ldots,\widetilde{\boldsymbol{B}}_{2m}{}^{\mathrm{T}},\ldots,\widetilde{\boldsymbol{B}}_{m1}{}^{\mathrm{T}},\ldots,\widetilde{\boldsymbol{B}}_{mm}{}^{\mathrm{T}}]^{\mathrm{T}},\\
\boldsymbol{C}=\mathrm{diag}\left\{\;\hat{\boldsymbol{C}}_1,\;\;\hat{\boldsymbol{C}}_2,\ldots,\;\;\hat{\boldsymbol{C}}_m\right\},\end{array} \tag{37a}$$

where

$$\hat{A}_i = \mathrm{diag}\left\{\widetilde{A}_1, \widetilde{A}_2, \ldots, \widetilde{A}_m\right\} \quad , \quad \hat{C}_i = \left[\widetilde{C}_1,\ \widetilde{C}_2, \ldots,\ \widetilde{C}_m\right] \quad i = 1,2,\ldots,m. \tag{37b}$$

Note that although notations of $x(n)$, $\boldsymbol{A}$,$\boldsymbol{B}$, and $\boldsymbol{C}$ are utilized both in Sections 3.1 and 4.1, the details are not the same as they represent models for systems with different types of time delays.

As the above state space realization is controllable and observable, the prediction of future model output value is obtained with

$$\begin{gathered}
\boldsymbol{T} = \mathrm{diag}\left\{\widetilde{\boldsymbol{T}}_1, \widetilde{\boldsymbol{T}}_2, \ldots, \widetilde{\boldsymbol{T}}_m\right\} \quad , \quad \widetilde{\boldsymbol{T}}_i = [\boldsymbol{T}_1\ , \boldsymbol{T}_2, \ldots, \boldsymbol{T}_m], \\
\boldsymbol{T}_j = \begin{bmatrix} \boldsymbol{C}_j \boldsymbol{A}_j^{d_i+1} \\ \boldsymbol{C}_j \boldsymbol{A}_j^{d_i+2} \\ \vdots \\ C_j A_j^{d_i+P_i} \end{bmatrix} = \left[\boldsymbol{0}, \boldsymbol{0}, \cdots, \boldsymbol{0}, \overline{\boldsymbol{T}}_{y_j}, \overline{\boldsymbol{T}}_{u_j}\right], \\
\overline{\boldsymbol{T}}_{y_j} = a_j \overline{\boldsymbol{T}}_{u_j} = \left[a_j\ , a_j^2, \ldots,\ a_j^{P_i}\right]^{\mathrm{T}} \quad j = 1,2,\ldots,m, \\
\boldsymbol{T}_y = [\boldsymbol{T}_{y_1}, \boldsymbol{T}_{y_2}, \ldots, \boldsymbol{T}_{y_m}] \quad , \quad \boldsymbol{T}_{y_1} = \begin{bmatrix} \widetilde{\boldsymbol{T}}_y \\ \\ \boldsymbol{0} \end{bmatrix}, \quad \boldsymbol{T}_{y_2} = \begin{bmatrix} \boldsymbol{0} \\ \widetilde{\boldsymbol{T}}_y \\ \boldsymbol{0} \end{bmatrix} \quad , \quad \boldsymbol{T}_{y_m} = \begin{bmatrix} \boldsymbol{0} \\ \\ \widetilde{\boldsymbol{T}}_y \end{bmatrix}, \quad \widetilde{\boldsymbol{T}}_y = [\overline{\boldsymbol{T}}_{y_1}, \overline{\boldsymbol{T}}_{y_2}, \ldots, \overline{\boldsymbol{T}}_{y_m}], \\
\boldsymbol{T}_u = [\boldsymbol{T}_{u_1}, \boldsymbol{T}_{u_2}, \ldots, \boldsymbol{T}_{u_m}] \quad , \quad \boldsymbol{T}_{u_1} = \begin{bmatrix} \widetilde{\boldsymbol{T}}_u \\ \\ \boldsymbol{0} \end{bmatrix}, \quad \boldsymbol{T}_{u_2} = \begin{bmatrix} \boldsymbol{0} \\ \widetilde{\boldsymbol{T}}_u \\ \boldsymbol{0} \end{bmatrix} \quad , \quad \boldsymbol{T}_{u_m} = \begin{bmatrix} \boldsymbol{0} \\ \\ \widetilde{\boldsymbol{T}}_u \end{bmatrix}, \quad \widetilde{\boldsymbol{T}}_u = [\overline{\boldsymbol{T}}_{u_1}, \overline{\boldsymbol{T}}_{u_2}, \ldots, \overline{\boldsymbol{T}}_{u_m}],
\end{gathered} \tag{38}$$

and $\boldsymbol{0}$ is a zero matrix with corresponding dimensions.

$$\boldsymbol{L} = [L_{ij}] \quad , \quad i,j = 1,2,\ldots,m, \tag{39a}$$

$$L_{ij} = k_{ij}(1-a_j)\left\{(1-b_j)\begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ a_j & 1 & \ddots & \vdots & \vdots \\ \vdots & a_j & \ddots & 0 & \vdots \\ \vdots & \vdots & \ddots & 1 & 0 \\ \vdots & \vdots & \cdots & a_j & 1 \\ \vdots & \vdots & \cdots & \vdots & 1+a_j \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ a_j^{P_i-1} & a_j^{P_i-2} & \cdots & a_j^{P_i-M_j+1} & 1+a_j+\cdots+a_j^{P_i-M_j} \end{bmatrix} + b_i \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \\ a_j & 1 & \ddots & \vdots & \vdots \\ \vdots & a_j & \ddots & 0 & \vdots \\ \vdots & \vdots & \ddots & 1 & 0 \\ \vdots & \vdots & \cdots & a_j & 1 \\ \vdots & \vdots & \cdots & \vdots & 1+a_j \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ a_j^{P_i-2} & a_j^{P_i-3} & \cdots & a_j^{P_i-M_j} & 1+a_j+a_j^{P_i-M_j-1} \end{bmatrix}\right\}, \quad i,j = 1,2,\ldots,m. \tag{39b}$$

Using Equations (2b), (34), and (35), we have

$$\overline{\boldsymbol{x}}_y(n) = \boldsymbol{J}(z)\boldsymbol{G}(z)^{-1}\boldsymbol{y}(n). \tag{40a}$$

where

$$\boldsymbol{J}(z) = \left[\boldsymbol{J}_1(z)^{\mathrm{T}}, \boldsymbol{J}_2(z)^{\mathrm{T}}, \ldots,\ \boldsymbol{J}_m(z)^{\mathrm{T}}\right]^{\mathrm{T}} \ , \ \boldsymbol{J}_i(z) = \mathrm{diag}\{\widetilde{g}_{i1}(z), \widetilde{g}_{i2}(z), \ldots, \widetilde{g}_{im}(z)\} \quad i = 1,2,\ldots,m \tag{40b}$$

Let

$$\begin{gathered}
\overline{\boldsymbol{T}}_u = \overline{\boldsymbol{T}}_y \boldsymbol{F}\,, \\
\boldsymbol{F} = \mathrm{diag}\left\{\widetilde{\boldsymbol{F}}_1, \widetilde{\boldsymbol{F}}_2, \ldots, \widetilde{\boldsymbol{F}}_m\right\} \ , \quad \widetilde{\boldsymbol{F}}_i = \mathrm{diag}\left\{a_1^{-1}, a_2^{-1}, \ldots, a_m^{-1}\right\} \ \ i = 1,2,\ldots,m,
\end{gathered} \tag{41}$$

then from Equation (41), we have

$$\boldsymbol{F}\overline{\boldsymbol{x}}_u(n) = \widetilde{\boldsymbol{E}}\boldsymbol{u}(n-1) = \widetilde{\boldsymbol{E}}\boldsymbol{Z}(z)\boldsymbol{u}(n), \tag{42a}$$

where

$$\begin{gathered} \widetilde{\boldsymbol{E}} = \left[\widetilde{\boldsymbol{E}}_1{}^{\mathrm{T}}, \widetilde{\boldsymbol{E}}_2{}^{\mathrm{T}}, \ldots, \widetilde{\boldsymbol{E}}_m{}^{\mathrm{T}}\right]^{\mathrm{T}}, \\ \widetilde{\boldsymbol{E}}_i = \mathrm{diag}\{k_{i1}(1-a_1)(b_1/a_1), k_{im}(1-a_2)(b_2/a_2), \ldots, k_{im}(1-a_m)(b_m/a_m)\} \quad i = 1,2,\ldots,m, \\ \boldsymbol{Z}(z) = \mathrm{diag}\{\underbrace{z^{-1}, z^{-1}, \ldots, z^{-1}}_{m}\}. \end{gathered} \tag{42b}$$

Hence, the optimal solution in Equation (15a) is transformed into

$$\boldsymbol{u}(n) = \boldsymbol{\Gamma}(\boldsymbol{w}(n) - \boldsymbol{b}(n)) - \boldsymbol{\Omega}(\boldsymbol{J}(z)\boldsymbol{G}(z)^{-1}\boldsymbol{y}(n) + \widetilde{\boldsymbol{E}}\boldsymbol{Z}(z)\boldsymbol{u}(n)), \tag{43}$$

where

$$\boldsymbol{\Gamma} = \left[\gamma_{ij}\right], \boldsymbol{\Omega} = [\boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2, \ldots, \boldsymbol{\Omega}_{\boldsymbol{m}}], \boldsymbol{\Omega}_i = \left[\delta_{ij}\right], \quad i, j = 1,2,\ldots,m. \tag{44}$$

4.2. Closed-Loop Analysis

Let $b(n) = 0$, so the plant and model outputs are the same. Using Equation (43), we have

$$(\boldsymbol{I} + \boldsymbol{\Omega}\boldsymbol{J}(z) + \boldsymbol{\Omega}\widetilde{\boldsymbol{E}}\boldsymbol{Z}(z))\boldsymbol{G}(z)^{-1}\boldsymbol{y}(n) = \boldsymbol{\Gamma}\boldsymbol{w}(n). \tag{45}$$

It can be shown that

$$\boldsymbol{\Gamma} = (\boldsymbol{I} + \boldsymbol{\Omega}\boldsymbol{J}(1) + \boldsymbol{\Omega}\widetilde{\boldsymbol{E}})\boldsymbol{K}^{-1}. \tag{46}$$

Hence, the closed-loop model is obtained as

$$\boldsymbol{y}_p(n) = \boldsymbol{G}_{cl}(z)\boldsymbol{w}(n), \tag{47a}$$

where

$$\boldsymbol{G}_{cl}(z) = \boldsymbol{G}(z)(\boldsymbol{I} + \boldsymbol{\Omega}\boldsymbol{J}(z) + \boldsymbol{\Omega}\widetilde{\boldsymbol{E}}\boldsymbol{Z}(z))^{-1}(\boldsymbol{I} + \boldsymbol{\Omega}\boldsymbol{J}(1) + \boldsymbol{\Omega}\widetilde{\boldsymbol{E}})\boldsymbol{K}^{-1}. \tag{47b}$$

To further study the closed-loop behavior of the system, the gain matrix $\boldsymbol{\Omega}$ is designed as

$$\boldsymbol{\Omega}_i = \mathrm{diag}\{\delta_1, \delta_2, \cdots, \delta_m\}. \tag{48}$$

Then, the following characteristic polynomials are derived

$$\begin{gathered} z - a_1 + \delta_1 \sum_{i=1}^{m} k_{i1}(1-a_1)(1 - b_1 + b_1 a_1{}^{-1}) = 0, \\ z - a_2 + \delta_2 \sum_{i=1}^{m} k_{i2}(1-a_2)(1 - b_2 + b_2 a_2{}^{-1}) = 0, \\ \vdots \\ z - a_m + \delta_m \sum_{i=1}^{m} k_{m1}(1-a_m)(1 - b_m + b_m a_m{}^{-1}) = 0. \end{gathered} \tag{49}$$

Therefore, we can do the pole placement for desired performance by choosing the proper gain value δ_i from gain matrix $\boldsymbol{\Omega}$ in each equation of Equation (49).

4.3. Tuning Method for Desired Performance

Assuming the control horizon is chosen as one, we derive the parameter tuning formulas that achieve the desired performance, and the details are summarized in the following theorem.

Theorem 3. *On the condition of no active constraints and no model mismatch, the achievable gain matrix corresponding to the desired closed-loop performance for MPC tuning problem should satisfy the following inequalities:*

$$\begin{array}{c} \boldsymbol{Q} > 0, \\ \boldsymbol{\Omega}_{di} < \left(\boldsymbol{L}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{L}\right)^{-1}\boldsymbol{L}^{\mathrm{T}}\boldsymbol{Q}\overline{\boldsymbol{T}}_{y_1} \qquad i = 1, 2, \ldots, m, \end{array} \tag{50}$$

and $\boldsymbol{\Omega}_d$ must also satisfy the closed-loop stability condition by the Jury criterion [18]. Through choosing $\boldsymbol{R}$ and $\boldsymbol{Q}$ as the tuning parameters, the tuning formulas for the desired gain matrix are obtained from

$$\begin{array}{c} \boldsymbol{L}^{\mathrm{T}}\boldsymbol{Q}[\overline{\boldsymbol{T}}_{y_1}\boldsymbol{\Omega}_{d1}{}^{-1}\boldsymbol{\Omega}_{di} - \overline{\boldsymbol{T}}_{y_i}] = 0 \qquad i = 1, 2, \ldots, m, \\ \boldsymbol{R} = L^{\mathrm{T}}\boldsymbol{Q}[\overline{\boldsymbol{T}}_{y_1}\boldsymbol{\Omega}_{d1}{}^{-1} - \boldsymbol{L}]. \end{array} \tag{51}$$

Proof. Under the condition of no active constraint, the tuning formulas are obtained as

$$[\boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2, \ldots, \boldsymbol{\Omega}_m] = \left(\boldsymbol{R} + \boldsymbol{L}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{L}\right)^{-1}\boldsymbol{L}^{\mathrm{T}}\boldsymbol{Q}\overline{\boldsymbol{T}}_y. \tag{52}$$

From Equations (38) and (52), we have

$$\begin{array}{c} (\boldsymbol{R} + \boldsymbol{L}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{L})\boldsymbol{\Omega}_1 = \boldsymbol{L}^{\mathrm{T}}\boldsymbol{Q}\overline{\boldsymbol{T}}_{y_1}, \\ (\boldsymbol{R} + \boldsymbol{L}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{L})\boldsymbol{\Omega}_2 = \boldsymbol{L}^{\mathrm{T}}\boldsymbol{Q}\overline{\boldsymbol{T}}_{y_2}, \\ \vdots \\ (\boldsymbol{R} + \boldsymbol{L}^{\mathrm{T}}\boldsymbol{Q}L)\boldsymbol{\Omega}_m = \boldsymbol{L}^{\mathrm{T}}\boldsymbol{Q}\overline{\boldsymbol{T}}_{y_m}. \end{array} \tag{53}$$

Then, the proof follows from Equation (53) and $\boldsymbol{R}$ and $\boldsymbol{Q}$ are all positive. □

It should be noted that the analytical solution of $\boldsymbol{Q}$ cannot be obtained directly. It can be achieved via the following optimization problem

$$\begin{array}{l} \min\limits_{Q} \sum\limits_{i=2}^{m} \|\boldsymbol{L}^{\mathrm{T}}\boldsymbol{Q}[\overline{\boldsymbol{T}}_{y_1}\boldsymbol{\Omega}_{d1}{}^{-1}\boldsymbol{\Omega}_{di} - \overline{\boldsymbol{T}}_{y_i}]\|, \\ s.t. \quad \left(\boldsymbol{L}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{L}\right)^{-1}\boldsymbol{L}^{\mathrm{T}}\boldsymbol{Q}\overline{\boldsymbol{T}}_{y_i} - \boldsymbol{\Omega}_{di} > 0, \\ \qquad\qquad \boldsymbol{Q} > 0, \end{array} \tag{54}$$

Then for $\boldsymbol{R}$ we obtain the exact solution as

$$\boldsymbol{R} = \boldsymbol{L}^{\mathrm{T}}\boldsymbol{Q}[\overline{\boldsymbol{T}}_{y_1}\boldsymbol{\Omega}_{d1}{}^{-1} - \boldsymbol{L}]. \tag{55}$$

Note that although the proposed method is developed based on unconstrained MPC, it can be applied to constrained systems (minor modifications may be necessary). More specifically, given a real industrial system, two different scenarios normally exist: (1) the constraints are relatively loose: the MPC constraints of some industrial applications (e.g., paper-making processes) are utilized to avoid extreme situations [19,20], and therefore, are normally quite loose in contrast to the real control inputs obtained from solving the MPC problem. In this case, the control signals obtained from the unconstrained and constrained MPCs are almost the same, and therefore the proposed method can be directly applied; (2) the constraints are relatively tight: as the control signals from the constrained and unconstrained MPCs are normally not the same (especially for signals near the constraints), the performance of the proposed method may be affected, but a fine tune procedure can be incorporated to guarantee the control performance [20]. More specifically, the proposed method is first performed to obtain a set of tuning parameters; then, consider the obtained tuning results as the starting point; and perform a fine tune around the obtained controller parameters based on trial-and-error or intelligent algorithms (e.g., particle swarm optimization) to search for the desired performance. It is worthwhile mentioning that as the control signal of a constrained MPC is obtained via solving a constrained QP (Quadratic Programming) online, the explicit solution to the MPC problem may not exist, and therefore the theoretical analyses in MPC tuning algorithms are normally constructed via the closed-loop transfer function derived based on unconstrained MPC [14,15,19,20]. Note that for MPC tuning of MIMO FOPFDT systems, the existing methods normally approximate the fractional order dead times to integral values to facilitate the theoretically analysis, which, however, may lead to performance degradation and even stability problem. The method proposed in this work directly considers the fractional order property of the dead times in the closed-loop analysis to improve the MPC performance, which could extend MPC method to more and more relevant industrial processes.

It is worth noting that if the considered industrial systems are without constraints, the control performance of the unconstrained MPC tuned via the proposed method and other optimization based control strategy (e.g., LQR (Linear Quadratic Regulator) method) could be close if the weighting matrices of the latter approach are also well tuned via, for example, the pole placement method. However, as the intention of the paper is to provide some useful tuning techniques for MPC controllers such that the industrial MPC systems (especially MIMO FOPFDT systems) can be tuned more efficiently and effectively, the simulation example is provided for a constrained MPC systems, rather than the comparison with other classical MIMO control strategies.

In addition, the proposed method may also to be applied for industrial applications with non-perfect models, but the following modifications need to be made: first, the small-gain theorem should be utilized instead of Jury's criterion for stability analysis. More specifically, given the model-plant mismatch specified via either unstructured or parametric uncertainty, the region of the tuning parameters to guarantee the stability needs to be calculated, and this can be obtained by checking the small gain condition based on the closed-loop transfer function developed in this work; second, rather than the standard time domain indices (e.g., settling time), the robust time domain performance indices [20] need to be employed to better characterize the desired performance of uncertain systems; to perform MPC tuning towards such indices, some existing results in robust performance analysis can be utilized. For example, the robust tuning may be performed based on the property that the worst-case behavior of an uncertain system controlled by MPC can normally be evaluated based on extreme uncertain systems, the model parameters of which are selected at the boundaries of the uncertainty specifications [19,20].

5. Simulation Results

Consider the transfer function from [21] as

$$\boldsymbol{G}(s) = \begin{bmatrix} \frac{16.9}{s+5}e^{-0.003} & \frac{36.12}{s+11}e^{-0.003} \\ \frac{-9.57}{s+5}e^{-0.003} & \frac{-4.175}{s+11}e^{-0.003} \end{bmatrix}$$

This is a two-input two-output model of a turbo generator. The throttle valve position and excitation control are the inputs. The generator terminal voltage and generator load angle are the outputs. After discretizing the system by a sampling time of $T_s = 0.01$ s, we have $k_{11} = 3.38$, $k_{12} = 3.2836$, $k_{21} = -1.914$, $k_{22} = -0.3791$, $a_1 = 0.9512$, $a_2 = 0.8958$, $b_1 = b_2 = 0.3$, $d_1 = d_2 = 0$. The desired settling time for two outputs are less than 0.2 s and 0.2 s, and then the desired closed-loop poles are 0.8 and 0.8. Let $\boldsymbol{Q} = \boldsymbol{I}$. According to Equation (28b), we choose $\boldsymbol{\Omega}_d = \begin{bmatrix} 3.0514 & \\ & 0.8884 \end{bmatrix}$ and $P_1 = P_2 = 7$ satisfying Equation (30). Equation (31) gives $\boldsymbol{R} = \begin{bmatrix} 0.8518 & 3.6397 \\ 1.3944 & 6.7545 \end{bmatrix}$. Let the input constraints be $|u_1(n)| \leq 2.5$, $|u_2(n)| \leq 1.5$, and the output constraints be $0.8 \leq y_1(n) \leq 2.2$, $|y_2(n)| \leq 0.6$. Figure 1 shows that all the outputs are able to track the references accurately with inactive constraints.

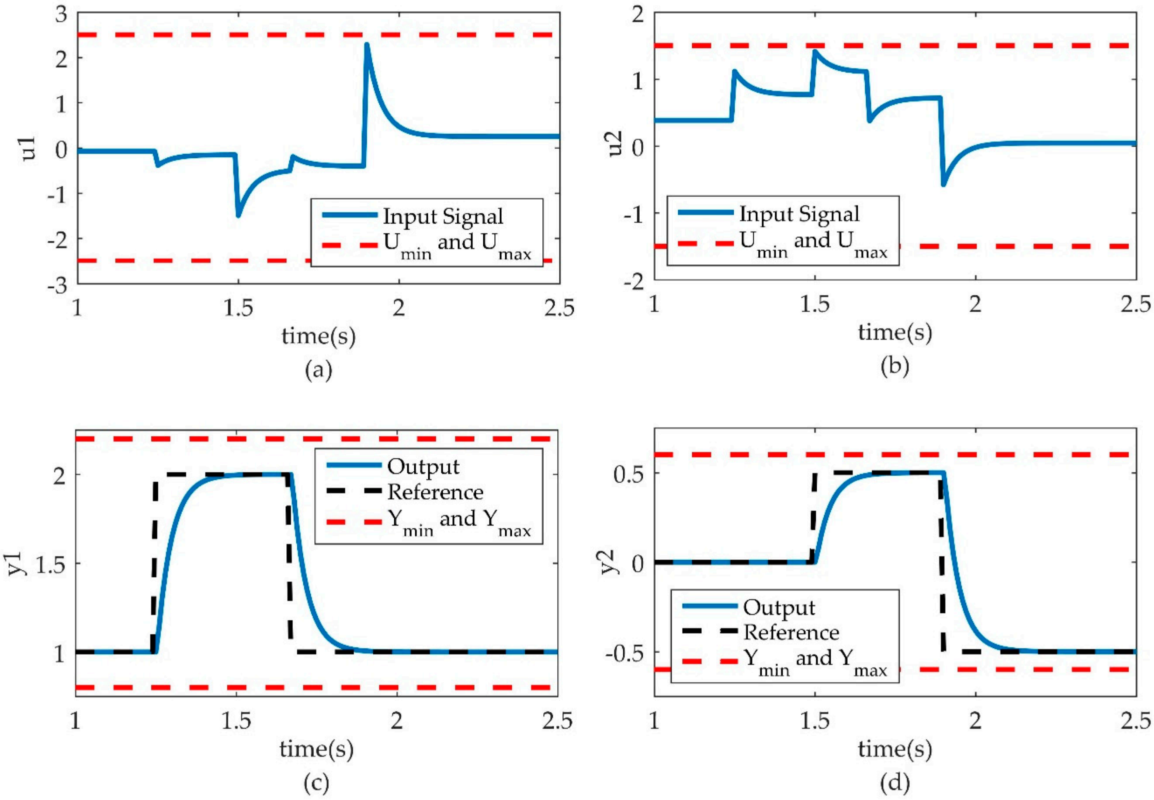

Figure 1. Closed-loop response for the same dead times. (**a**) The first control signal; (**b**) the second control signal; (**c**) the generator terminal voltage output; (**d**) the generator load angel output.

Considering the dead times are different, we assume $b_1 = 0.3$, $b_2 = 0.6$. The desired settling time for two outputs is less than 0.2 s and 0.2 s, and then the desired closed-loop poles are also 0.8 and 0.8. Therefore, Equation (49) leads to $[\mathbf{\Omega}_1 \quad \mathbf{\Omega}_2] = \begin{bmatrix} 2.0814 & 0 & 2.0814 & 0 \\ 0 & 0.2959 & 0 & 0.2959 \end{bmatrix}$. Assuming $P = 4$, we can obtain $\boldsymbol{Q}$ and $\boldsymbol{R}$ from Equation (53) as

$$\boldsymbol{Q} = \begin{bmatrix} 0.9028 & 0.1830 & -0.03344 & -0.0674 & 0.3049 & 0.2118 & 0.1114 & -0.0565 \\ 0.1830 & 0.7841 & -0.0815 & -0.1062 & 0.0813 & 0.0787 & 0.1590 & 0.0836 \\ -0.0334 & -0.0815 & 0.7315 & -0.0396 & -0.0621 & 0.1782 & 0.1886 & 0.2421 \\ -0.0674 & -0.1062 & -0.0396 & 0.7924 & -0.0324 & 0.1367 & 0.3039 & 0.4041 \\ 0.3049 & 0.0813 & -0.0621 & -0.0324 & 0.8720 & 0.1334 & -0.0564 & -0.2034 \\ 0.2118 & 0.0787 & 0.1782 & 0.1367 & 0.1334 & 0.7679 & 0.0527 & -0.1039 \\ 0.1114 & 0.1590 & 0.1886 & 0.3039 & -0.0564 & 0.0527 & 0.7504 & -0.0331 \\ -0.0565 & 0.0836 & 0.2421 & 0.4041 & -0.2034 & -0.1039 & -0.0331 & 0.7455 \end{bmatrix}, \boldsymbol{R} = \begin{bmatrix} 0.0860 & 0.6772 \\ 0.2858 & 2.2627 \end{bmatrix}.$$

Let the input constraints be $|u_1(n)| \le 1.6$, $|u_2(n)| \le 1.3$, and the output constraints be $0.8 \le y_1(n) \le 2.2$, $|y_2(n)| \le 0.6$. Figure 2 gives the closed-loop responses for different dead time cases with and without MPC constraints.

The results above show the effectiveness of the proposed method.

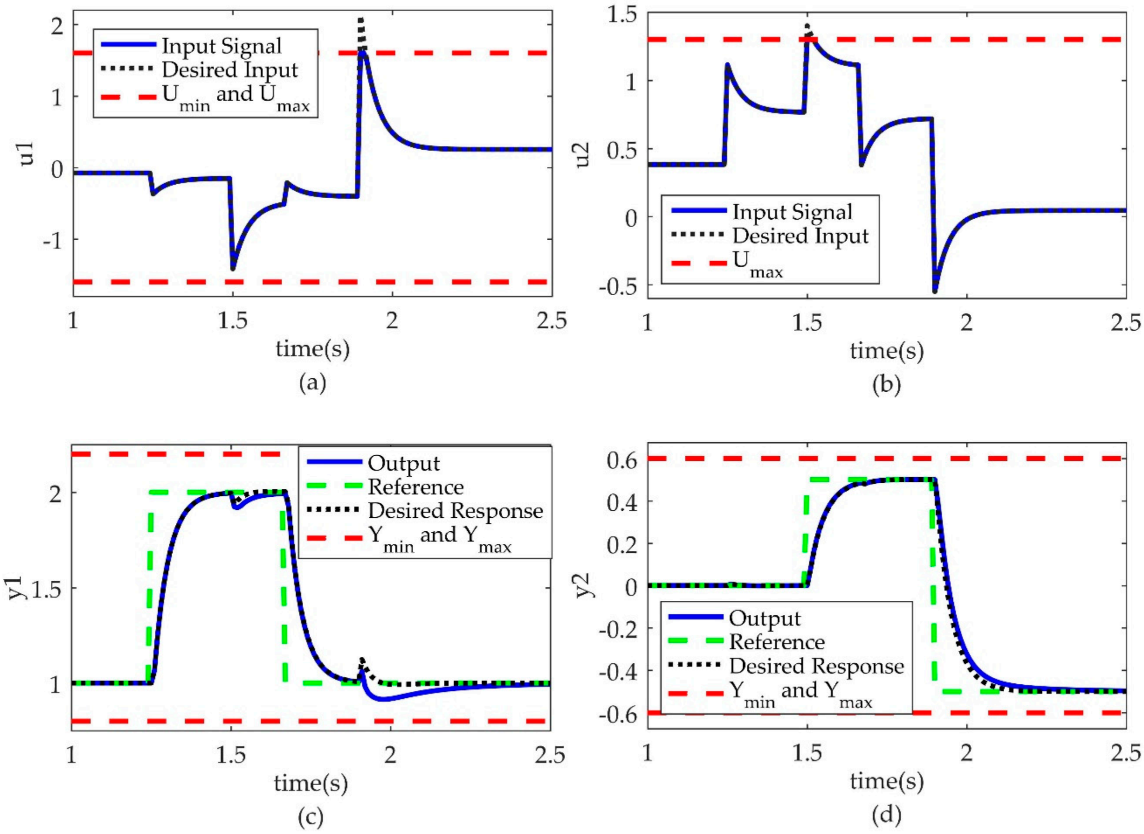

Figure 2. Closed-loop responses for different dead time cases. (**a**) The first control signal; (**b**) the second control signal; (**c**) the generator terminal voltage output; (**d**) the generator load angel output.

6. Conclusions

In this paper, the MPC tuning problem of MIMO FOPFDT system is addressed under the condition that the constraints are inactive and the model mismatch does not exist. The closed-loop transfer functions of these systems are provided and the decoupling condition is given. The analytical tuning formulas guaranteeing the closed-loop stability and desired performance are derived for control horizon of one. Finally, simulation shows the effectiveness of the proposed method. The results of this paper lay a good foundation for further study on the analytical MPC tuning method for general multivariable FOPDT with fractional dead time systems.

Author Contributions: Conceptualization, methodology, and writing: N.H.; software and validation: Y.J.; supervision: L.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China, grant number 61903291; this research was also funded by Natural Science Foundation of Shaanxi Province, grant number 2019JQ-004.

Acknowledgments: The authors would like to thank the editor and reviewers for their constructive suggestions that have improved the quality of the paper.

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

References

1. Rawlings, J.B.; Mayne, D.Q. *Model Predictive Control: Theory and Design*, 5th ed.; Nob Hill Publishing: Madison, WI, USA, 2015; pp. 1–8.
2. Moksadur, R.; Anders, A.; Konstantinos, K. An approach for feedforward model predictive control of continuous pulp digesters. *Processes* **2019**, *7*, 602–622.
3. Abhinav, G.; Hassan, A.A.; Prashant, M.; Michael, R.T. Handling constraints and raw material variability in rotomolding through data-driven model predictive control. *Processes* **2019**, *7*, 610–624.

4. Darby, M.L.; Nikolaou, M. MPC: Current practice and challenges. *Control Eng. Pract.* **2012**, *20*, 328–342. [CrossRef]
5. Garriga, J.L.; Soroush, M. Model predictive control tuning methods: A review. *Ind. Eng. Chem. Res.* **2010**, *49*, 3505–3515. [CrossRef]
6. Lee, J.H. Model predictive control: Review of the three decades of development. *Int. J. Control Autom.* **2015**, *9*, 415–424. [CrossRef]
7. Xi, Y.G.; Li, D.W.; Lin, S. Model predictive control—Status and challenges. *Acta Autom. Sin.* **2013**, *39*, 222–236. [CrossRef]
8. Scokaert, P.; Rawlings, J.B. Constrained linear quadratic regulation. *IEEE Trans. Autom. Control* **1998**, *43*, 1163–1169. [CrossRef]
9. Shridhar, R.; Cooper, D.J. A tuning strategy for unconstrained multivariable model predictive control. *Ind. Eng. Chem. Res.* **1998**, *37*, 4003–4016. [CrossRef]
10. Nery Júnior, G.A.; Martins Márcio, A.F.; Kalid, R. A PSO-based optimal tuning strategy for constrained multivariable predictive controllers with model uncertainty. *ISA Trans.* **2014**, *53*, 560–567. [CrossRef] [PubMed]
11. Bagheri, P.; Khakisedigh, A. An ANOVA based analytical dynamic matrix controller tuning procedure for FOPDT models. *AUT J. Modeling Simul.* **2010**, *2*, 55–64.
12. Shah, G.; Engell, S. Tuning MPC for desired closed-loop performance for MIMO systems. In Proceedings of the American Control Conference, San Francisco, CA, USA, 29 June–1 July 2011; pp. 4404–4409.
13. Yong, K.H.; Farouq, S.M.; Hak, K.Y. Generalized predictive control with dual adaptation. *Chem. Eng. Sci.* **2012**, *84*, 479–493.
14. Bagheri, P.B.; Khakisedigh, A. Analytical approach to tuning of model predictive control for first-order plus dead time models. *IET Control Theory Appl.* **2013**, *7*, 1806–1817. [CrossRef]
15. Bagheri, P.B.; Khakisedigh, A. An analytical tuning approach to multivariable model predictive controllers. *J. Process Control* **2014**, *24*, 41–54. [CrossRef]
16. Bagheri, P.B.; KhakiSedigh, A. Closed form tuning equations for model predictive control of first-order plus fractional dead time models. *Int. J. Control Autom. Syst.* **2015**, *13*, 73–80. [CrossRef]
17. Sobh, T.; Elleithy, K. *Emerging Trends in Computing, Informatics, Systems Sciences, and Engineering*; Springer Science & Business Media: New York, NY, USA, 2013; pp. 147–158.
18. Hu, S.S. *The Principle of Automatic Control*, 5th ed.; Science Press: Beijing, China, 2007; pp. 352–354.
19. Fan, J.; Stewart, G. Automated tuning of large-scale multivariable model predictive controllers for spatially-distributed processes. In Proceedings of the American Control Conference, Minneapolis, MN, USA, 14–16 June 2006; pp. 5517–5523.
20. He, N.; Liu, X.; Forbes, M.; Backstrom, J.; Chen, T. Robust tuning of cross-directional model predictive controllers for paper-making processes. *IEEE Trans. Control Syst. Technol.* **2018**, *13*, 884–895. [CrossRef]
21. Maciejowski, J.M. *Multivariable Feedback Design*; Addison Wesley: Wokingham, UK, 1989; pp. 166–167.