

# 杭州电子科技大学

## 硕士学位论文

题目：分数阶系统的动态矩阵控制算法研究

研究生 汪大卫

专业 控制工程

指导教师 张日东 研究员

邹洪波 副教授

完成日期 2017 年 3 月

杭州电子科技大学硕士学位论文

分数阶系统的动态矩阵控制算法研究

研 究 生： 汪大卫

指导教师： 张日东 研究员

邹洪波 副教授

2017 年 3 月

**Dissertation Submitted to Hangzhou Dianzi  
University  
for the Degree of Master**

**Research on Dynamic Matrix Control  
Algorithm for Fractional Order Systems**

**Candidate:** Wang Da-wei  
**Supervisors:** Prof. Zhang Ri-dong  
Prof. Zou Hong-bo

March, 2017

# 杭州电子科技大学

## 学位论文原创性声明和使用授权说明

### 原创性声明

本人郑重声明： 所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不含任何其他个人或集体已经发表或撰写过的作品或成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。

申请学位论文与资料若有不实之处，本人承担一切相关责任。

论文作者签名： 日期： 年 月 日

### 学位论文使用授权说明

本人完全了解杭州电子科技大学关于保留和使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属杭州电子科技大学。本人保证毕业离校后，发表论文或使用论文工作成果时署各单位仍然为杭州电子科技大学。学校有权保留送交论文的复印件，允许查阅和借阅论文；学校可以公布论文的全部或部分内容，可以允许采用影印、缩印或其它复制手段保存论文。（保密论文在解密后遵守此规定）

论文作者签名： 日期： 年 月 日

指导教师签名： 日期： 年 月 日

## 摘 要

分数阶微积分,是将微积分的阶次从常规的整数域推广至实数域甚至是复数范围所得。现实中的动态系统大多为分数阶的,整数阶微分方程描述的模型往往只能体现控制系统的部分动态特性,而分数阶微积分的任意阶次数及其特有的记忆特性,使得分数阶微分方程恰能更精确的表征系统的内在特性。因而研究分数阶系统的控制问题具有极其重要的理论意义和工程价值。

动态矩阵控制(DMC)作为预测控制中的代表性算法,具有对于模型要求低、鲁棒性强、处理时延简单易行等优点,在强耦合、大时滞及无法建立精确数学模型的过程中十分适用。然而该方法同时也存在在线计算量大、采样周期不宜过小等局限,从而使得其在模型阶次、环境扰动及非线性等方面存在不确定性较大的控制系统中往往难以达到预期的控制效果。因此本文对传统的动态矩阵控制方法进行改进,并将其运用于分数阶系统的控制中。

本文的主要工作可以分为三个部分:第一部分为:对两种改进的动态矩阵控制方法进行介绍,并通过稳定汽油蒸汽压力控制的仿真实例与传统的动态矩阵控制进行对比。仿真结果表明基于PID控制结构改进的DMC方法在整体性能上的优势。第二部分为:将PID型动态矩阵控制方法运用于一类单入单出分数阶线性系统中,并通过加热炉温度控制的仿真实例中验证了该方法的控制性能。第三部分:考虑一类多变量分数阶系统,通过在分布式控制结构中引入PID型动态矩阵控制方法,得到一种改进的分布式动态矩阵控制方法,并将其用于一类多变量分数阶系统的控制中,最后通过MATLAB仿真研究验证了所提方法的可行性和有效性。

**关键词:** 分数阶微积分, 分数阶系统, 动态矩阵控制, PID 控制, 分布式动态矩阵控制

## ABSTRACT

The fractional order calculus is a generalization of the order of calculus from the conventional integer domain to the real number field and even the complex range. Most dynamic systems in practice are of fractional order and the integer order differential equation model can only describe part of dynamic characteristics of the control system. However, the inherent characteristics of these systems can be characterized by the fractional order differential equations more precisely because of the number of arbitrary orders and the distinctive memory feature of fractional order calculus. So it is of great theoretical significance and engineering value to study the control problem of fractional order system.

As a representative algorithm in the predictive control, there are low requirements for the model, strong robustness, simple processing delay in dynamic matrix control (DMC), and it is suitable in the process with large time delay and strong coupling, also in which cannot establish accurate mathematical model. However, there are some limitations in this method, such as the enormous online computation, the sampling period which should not be too small and so on, which makes it difficult to achieve the desired control effect in the control system with uncertainty in the model order, environmental disturbance and nonlinear etc.. In this paper, the traditional dynamic matrix control method is improved, and it is applied to the control of fractional order system.

The main work of this thesis can be divided into three parts: the first part serves as an introduction to two kinds of improved dynamic matrix control methods and compares them with the traditional dynamic matrix control in the simulation examples of stable gasoline vapor pressure control, which shows the advantages of the improved DMC method based on the PID control structure on the overall performance. In the second part, the PID-type dynamic matrix control method is applied in a class of single input and single output fractional order linear systems, and the control performance of the method is verified by the simulation examples of the temperature control of the heating furnace. The last part is that since the PID-type dynamic matrix control is introduced to the distributed control structure, an improved distributed dynamic matrix control method is obtained considering a class of multivariable fractional order systems. And it is utilized in the control of a class of multivariable

fractional order systems. Finally, the feasibility and effectiveness of the proposed method are verified by MATLAB simulation.

**Keywords:** fractional order calculus, fractional order system, dynamic matrix control , PID control, distributed dynamic matrix control

## 目 录

摘 要.....	I
<b>ABSTRACT.....</b>	<b>II</b>
目 录.....	IV
第 1 章 绪论.....	1
1.1 研究背景及意义.....	1
1.2 分数阶系统控制理论的研究现状.....	1
1.2.1 分数阶系统近似的研究现状.....	1
1.2.2 分数阶控制器的研究现状.....	2
1.3 模型预测控制的研究现状.....	3
1.4 分布式模型预测控制的研究现状.....	5
1.5 论文的主要工作及内容安排.....	6
第 2 章 理论基础.....	8
2.1 分数阶微积分.....	8
2.1.1 分数阶微积分的定义与性质.....	8
2.1.2 分数阶系统的描述.....	10
2.2 动态矩阵控制.....	11
2.2.1 动态矩阵控制的算法实现.....	11
2.2.2 动态矩阵控制方法的参数设计.....	14
2.3 分布式模型预测控制.....	17
2.3.1 分布式预测控制算法.....	17
2.3.2 分布式动态矩阵控制.....	19
2.4 本章小结.....	21
第 3 章 改进的动态矩阵控制算法.....	22
3.1 引言.....	22
3.2 动态矩阵控制方法的改进.....	22
3.2.1 动态矩阵控制中预测模型的建立.....	22
3.2.2 动态矩阵控制性能指标的选取.....	24
3.3 案例研究：稳定汽油蒸汽压力控制系统.....	26



3.3.1 稳定系统的工艺流程.....	27
3.3.2 稳定系统的控制目标.....	28
3.3.3 改进的 DMC 方法与传统 DMC 方法的比较 .....	28
3.4 本章小结 .....	31
第 4 章 基于分数阶系统的动态矩阵控制.....	32
4.1 引言 .....	32
4.2 分数阶系统的 PID 型动态矩阵控制方法 .....	32
4.2.1 分数阶模型的处理.....	32
4.2.2 分数阶 PID 型动态矩阵控制器的设计 .....	33
4.3 案例研究：加热炉温度控制系统.....	35
4.3.1 加热炉工艺流程介绍.....	35
4.3.2 模型对比与分析.....	36
4.3.3 仿真结果及分析.....	38
4.4 本章小结 .....	40
第 5 章 基于分数阶系统的分布式动态矩阵控制.....	41
5.1 引言 .....	41
5.2 分数阶系统的分布式动态矩阵控制方法 .....	41
5.2.1 分布式动态矩阵控制中预测模型的计算.....	41
5.2.2 分布式动态矩阵控制性能指标的选取.....	44
5.2.3 分数阶分布式动态矩阵控制器的设计.....	45
5.3 仿真研究 .....	46
5.4 本章小结 .....	51
第 6 章 总结与展望.....	52
6.1 本文总结 .....	52
6.2 研究展望 .....	52
致谢.....	54
参考文献.....	55
附录.....	61

## 第 1 章 绪论

### 1.1 研究背景及意义

模型预测控制(Model Predictive Control, MPC),简称为预测控制,是起源于上个世纪七十年代的一类新型先进控制算法,是继传统 PID 控制后在工业过程中应用最为广泛且有效的控制算法,并已成为处理多变量、有约束的复杂工业过程控制优化问题的公认标准<sup>[1-3]</sup>。预测控制继承了现代控制理论中的优化思想,并通过有限时域的不断在线优化,即所谓的滚动优化思想,替代了传统最优控制理论中一次性的全局优化,且在滚动优化的每一时刻通过实时信息对模型输出进行反馈校正,一定程度上克服了对最小化参数模型进行辨识的困难,同时使得在线优化的实时计算规模大幅降低,并加强了控制系统的鲁棒性,符合实际工业中对过程控制的具体要求。三十多年来,预测控制在炼油、化工及电力等各行业的过程控制系统中的成功应用,已使其成为最有成效和应用潜力的先进过程控制方法<sup>[4]</sup>。

随着实际工业过程中对产品质量和安全操作等要求的不断提高,对象的内在复杂特性及外在环境成为实际问题中必须考虑的因素,这在一定程度上给传统的过程控制带来了诸多挑战。因此为实际过程中的复杂系统建立处理简单且精度较高的数学模型的需求十分迫切,而分数阶次的微分方程模型的出现则为这一需求提供了有效的途径。不同于整数阶次的微分方程所描述模型在表征复杂系统动态特性上的局限性,分数阶微积分的任意阶次及其特有的记忆特性恰能更加全面和精准的描述系统的内在特性,在实际过程中的适用范围更加广泛,因此研究分数阶系统的控制问题成为控制领域中又一项意义深远的研究课题。

### 1.2 分数阶系统控制理论的研究现状

一般情况下,将数学模型由分数阶微分方程描述的系统称为分数阶系统,用来解决其控制问题的理论即为分数阶系统的控制理论。它不仅仅是分数阶微积分理论在控制领域的应用,同时也是对经典控制理论的延伸。下面将从分数阶系统的近似处理及分数阶控制器两个方面分别对分数阶系统控制理论的发展主干及研究现状进行简要介绍。

#### 1.2.1 分数阶系统近似的研究现状

由于用于描述分数阶系统的微分方程阶次为分数,无法直接运用整数阶领域的理论方法,因此需要将分数阶系统变换为常规的整数阶系统方可研究,变换的主要方法有离散化及有理函数近似化<sup>[5]</sup>。目前,对分数阶系统进行近似处理

的方法通常分为直接近似与间接近似两种，前者主要是采用  $Z$  变换将分数阶系统近似处理为离散形式的整数阶系统，而后者则是采用 Laplace 变换将分数阶系统近似处理为连续形式的整数阶系统。

分数阶系统的直接近似方法主要包括基于 Euler 算子的幂级数展开法<sup>[6]</sup>、基于 Tustin 操作算子的连分式展开法<sup>[7]</sup>等，其中后者在高频部分的误差通常比较大<sup>[8]</sup>。Alaoui 通过结合 Euler 算子和 Tustin 操作算子，得到了 Al-Alaoui 算子，进而得到基于 Al-Alaoui 算子的连分式展开法的近似方法<sup>[9-10]</sup>。还有的学者利用由 Muir 递归结合 Tustin 操作算子得到的近似方法、由 Tustin 操作算子结合连分式展开法的近似方法等。直接近似法采用的主要处理方法有级数展开法或连分式展开法等，前者通常需要考虑级数的收敛域问题。实际中由于该方法收敛速度一般比较慢，而收敛域也往往难以确定，使得该方法的应用受到局限。在分数阶系统的近似处理上，连分式展开法的收敛速度一般快于指数序列展开法，并可以在频域中运用，但是连分式展开法在实现精度及离散化模型对原分数阶系统稳定性的保证上仍然存在不足。

研究者们也想到利用信号处理领域中的工具来对分数阶系统进行近似处理。有的想到利用 FIR(finite impulse response)对分数阶系统进行近似处理，但是该方法由于其所得近似系统的模型阶次数过高而失去可行性。还有的学者则想到将 IIR(infinite impulse response)用于分数阶系统的近似处理<sup>[11]</sup>，该方法通过梯形规则和 Simpson 积分规则的有效结合实现近似，在权重系数的优化选取上有待进一步研究。近似处理过程也可以通过最优化稳定的有理函数实现近似拟合<sup>[12]</sup>。该方法不仅弥补了连分式展开法存在的不足，同时还具有更好的拟合效果。Carlson 通过在迭代根中引入牛顿法来实现对分数阶算子的近似处理。Matsuda 通过对数空间上的一组点来实现对分数阶系统的逼近。与此同时，A.Oustaloup<sup>[13]</sup>、A.Chareff 等在前人的基础上分别给出了各自的近似方法，并获得了较为理想的拟合效果。目前看来 Oustaloup 近似方法在拟合效果上的表现更为突出，尽管该方法在近似频段的两端拟合效果并不理想，但整体近似效果值得肯定，并且由于其在实际中的可操作性及在近似响应时间上的优势而运用广泛。赵春娜给出的一种改进近似方法，在近似效果和近似响应时间上均有所改善，并证明了该方法的可行性<sup>[14]</sup>。改进近似法通过提高近似频段两端的近似效果，提高了整体近似精度，成为目前最精确的间接近似法。

### 1.2.2 分数阶控制器的研究现状

分数阶控制器就是在控制器的设计过程中引入分数阶微积分并对其特性加以利用而得。其中以法国 Oustaloup 等学者提出的分数阶鲁棒控制器最为经典，依据其法文名首字母也被称作 CRONE 控制器，共经历了三代发展过程。三代之

间存在的共同特性是：利用分数阶特有的性质，控制器可以对被控对象动态环节产生的变化做出相应补偿，使得系统闭环响应所产生的超调量以期望值保持恒定不变，即使得系统不受外界扰动及动态变化影响。第一代 CRONE 控制器<sup>[15]</sup>将简单放大器作为被控对象，其增益系数的变化由分数阶积分器做出补偿，因此控制器的增益变化不受相角裕度变化的影响。第二代 CRONE 控制器<sup>[16]</sup>通常考虑被控对象为线性定常的，其静态增益的变化将受到补偿作用。在第三代 CRONE 控制器<sup>[17]</sup>中，分数阶阶次由实数域拓展至复数域，被控对象的放大增益和相位变化将在同一时间受到复数阶次数动态环节的补偿作用。随着 CRONE 控制器的不断发展，在很多实际工程中得到成功应用。Oustaloup 等将第一代 CRONE 控制器成功用于两自由度的机器手臂运动控制中，并且控制效果比非线性的解耦控制更好<sup>[18]</sup>。Yousfi 等使用的 CRONE 控制器使得车辆中常见的减振问题得以解决<sup>[19]</sup>。Sabatier 等利用 CRONE 控制器来控制多质量负载的转速<sup>[20]</sup>。Cervera 等使用 CRONE 控制器来设计 QFT 系统的鲁棒性能<sup>[21-22]</sup>。Altet 等通过 CRONE 控制器解决了一类多变量系统中的路径追踪问题<sup>[23]</sup>。

另一种经典的分数阶控制器是由 Podlubny 最早提出的  $PI^\lambda D^\mu$  控制器<sup>[24]</sup>，由于作为常规 PID 控制器的扩展延续了其在工业中形式简单及调整容易的优点，并且更加灵活有效，一经提出便引发了诸多关注。陈阳泉教授以连分式展开的方式处理系统的离散化，分别提出了分数阶的微分器及积分器在连续和离散时域中的具体实现，同时设计出数字形式的分数阶微分器<sup>[25]</sup>。严慧采用一种极点阶次的搜索方法对控制参数满足系统时域性能指标的  $PI^\lambda D^\mu$  控制器进行设计<sup>[26]</sup>。梁涛年基于分数阶系统的鲁棒性条件，设计出一种能够满足指定相位裕度及幅值裕度的  $PI^\lambda D^\mu$  控制器，可以有效克服分数阶系统在时间常数上的不确定性变化<sup>[27]</sup>。此外，分数阶控制器的控制参数设计也逐渐得到研究者的重视。Monje 等设计了一种新型分数阶  $PI^\lambda D^\mu$  控制器，并在实验中对所设计的方法进行验证，说明了传统的分数阶控制器主要是以性能指标中的数值运算作为参数的数值整定方法<sup>[28]</sup>。Valério 等采用经典的 Z-N 法(Ziegler-Nichols Rule)来整定分数阶  $PI^\lambda D^\mu$  控制器的控制参数，并给出部分控制参数满足的基本条件<sup>[29]</sup>。Rahimian 等利用整数阶模型对分数阶  $PI^\lambda D^\mu$  控制器进行近似处理，并将整数阶 PID 控制器的传统参数整定方法推广至分数阶 PID 控制器参数设计中<sup>[30]</sup>。Bouafoura 等基于正交基函数法设计出一种包含分数阶 PID 控制器的闭环控制系统，并根据该系统的目标函数求解分数阶 PID 控制器的控制参数<sup>[31]</sup>。

### 1.3 模型预测控制的研究现状

模型预测控制(Model Predictive Control, MPC)作为先进控制算法的一种，是过程控制领域中兴起的一类新型计算机控制算法，其出现最早可以追溯到上

个世纪 70 年代后期<sup>[32]</sup>。该算法对模型精度的要求较低<sup>[33-34]</sup>，在多变量、含时滞及带约束控制问题的处理上表现极为优越<sup>[35]</sup>，同时具有很好的跟踪性能及鲁棒性能<sup>[36-38]</sup>。一般情况下，模型预测控制亦可简称为预测控制，其中主要包括如下几类典型算法：

(1)模型预测启发式控制 (Model Predictive Heuristic Control, MPHIC)：该算法基于有限脉冲响应建模，在所有预测控制中提出时间最早。

(2)动态矩阵控制 (Dynamic Matrix Control, DMC)：该算法基于有限阶跃响应建模，且只对渐进稳定的线性对象适用。

(3)模型算法控制 (Model Algorithmic Control, MAC)：该算法的模型通过采集对象的脉冲响应数据建立，适用范围与 DMC 基本相同，即只可以控制渐近稳定的线性对象。

(4)广义预测控制 (Generalized Predictive Control, GPC)：该算法继承了自校正控制中的优点，主要适用于随机离散系统。

上述这些预测控制算法的建模过程并不复杂，只需要利用滚动优化思想对反馈校正后的模型在各个时刻的最优控制量进行不断求解，并将其作用于被控对象即可。鉴于以上算法本身与实际过程中需求的完美契合，使得很多公司为此类算法专门设计了相应的软件。例如 Honeywell 公司设计的 RMPCT 软件，DMC 公司设计的 DMC-Plus 软件，Setpoint 公司设计的 SMC-Idcom 软件等等<sup>[4]</sup>。这些软件不仅使得企业的成本大大降低，同时也为推动实际工业过程的发展创造了巨大价值。

实际工业中存在的大量非线性过程往往需要经过相关近似线性化处理，然而经过线性化处理后的模型所设计的控制器势必会对实际系统的控制效果造成一定影响。因此如何在非线性系统中有效的运用预测控制成为众多学者关注的研究课题。其中一类就是利用 T-S(Takagi-Sugeno)模糊方法对非线性系统建模，然后结合预测控制的思想来设计相应控制器。Skrjanc 等利用 T-S 模糊模型设计了一种预测函数控制器，并将其应用于热交换器的实例中<sup>[39]</sup>。Oblak 等采用 T-S 模糊模型来逼近非线性系统进而得到其线性模型，最后由该模型的状态空间形式设计出控制器<sup>[40]</sup>。Su 等人基于 T-S 模型设计了一种自适应预测函数控制器，并在 pH 值的过程控制实例中验证了其性能<sup>[41]</sup>。综上所述可以看出大多数学者对于非线性系统的处理方法都比较类似，即采用 T-S 模型对非线性系统进行描述，并结合预测控制的思想对所描述系统设计相应的控制器<sup>[42, 43]</sup>。另一类则是采用神经网络作为非线性系统的描述方式，然后基于非线性搜索或线性化等方法求取最优控制律。Tsai 等针对递归神经网络描述的预测模型，设计了一种含约束的多变量预测控制器<sup>[44]</sup>。薛福珍等人采用神经网络并结合先验知识对非线性系统的

模型进行描述,并在计算中利用遗传算法实施优化,最后通过悬吊系统的仿真实例验证了所提方法的可行性<sup>[45]</sup>。石屹然等利用 RBF 神经网络描述 AFR 的系统模型,提出了一种可行信赖域的 SQP 优化算法<sup>[46]</sup>。黄德先等利用小波神经网络描述的模型设计了一种带输入约束的区域预测控制方法<sup>[47]</sup>。除此之外,还有学者采用如 Wiener 模型、Hammerstein 模型等特殊非线性模型<sup>[51-53]</sup>和支持向量机<sup>[48-50]</sup>设计相应预测控制器等等。经过总结不难发现,上述方法都是先利用神经网络描述对象的非线性模型,然后直接通过所得模型设计相应控制器或结合预测控制思想完成相应控制器设计。

实际工业过程中往往存在着各种约束限制,因而约束预测控制的研究也引发了诸多学者的关注。执行器通常工作在一个有限的范围,例如:物理条件对阀门造成的限制,使得其开度受限,在实际过程中,必须将预测控制每个时刻所求得的最优控制输入限制在要求的开度范围内,否则优化求解所得最优控制律往往不能对阀门产生有效作用。朱齐丹等利用状态空间模型设计了一种约束预测控制算法,并将其应用于增压锅炉汽包水位的实例中,有效改进了控制系统的鲁棒性和稳定性,仿真结果证明了该方法的快速性<sup>[54]</sup>。王喜庆等将一种约束预测控制算法应用于电厂过热蒸汽温度控制系统中,使得过程的动态性能及抗干扰性能得到改善<sup>[55]</sup>。于洋等针对动态及稳定过程中的变量会出现饱和的现象,通过调整性能指标,提出了一种带终端约束的预测控制方法,最后在 CSTR 系统中验证了该方法的可行性<sup>[56]</sup>。张晓颖等提出了一种含前馈-反馈结构的约束预测控制方法,并将其运用于六自由度机械臂的实时控制<sup>[57]</sup>等等。综上所述,尽管针对约束关系处理所提求解方法有不少,但是通常要综合考虑实际过程中对约束求解的快速性及有效性方可选择合适的方法<sup>[58-60]</sup>。

## 1.4 分布式模型预测控制的研究现状

分布式模型预测控制(也可简称为分布式预测控制)最早出现在21世纪初期,主要是为了解决一类大规模预测控制的在线优化问题而提出的。分布式预测控制的主要思路是借鉴纳什优化的理论和思想,将一个大规模复杂系统的在线优化问题分散到各个小规模子系统中分布式求解,因此系统的全局最优解与各子系统的优化解息息相关。与分散式控制不同的是,分布式预测控制通过在每个子系统实施优化的过程中与其他子系统进行信息交互,有效解决了子系统间存在的耦合问题。这样一来不仅使得系统控制规模的复杂性大大降低,同时弥补了分散式控制在应用于内部存在耦合关系的控制系统中的不足<sup>[61]</sup>。

分布式预测控制的提出引来了国内外诸多学者的关注,掀起了过程控制界的一波研究热潮。Kayacan等通过使用一种快速分布式非线性模型预测控制算法解决了一个自治拖拉机挂车系统的轨迹跟踪问题<sup>[62]</sup>。Stewart等采用分布式控制

思想完成非线性系统的优化控制任务,设计了一种满足其稳定性条件的分布式预测控制器<sup>[63]</sup>。Li等针对一阶的多智能体系统,研究了如何利用相邻信息解决基于分布式滚动时域控制(RHC)的一致性问题<sup>[64]</sup>。Farina等针对线性离散时间系统提出了一种新型的分布式预测控制(DPC)算法<sup>[65]</sup>。Perizzato等将分布式控制方案应用于小型独轮车车辆的编队控制中<sup>[66]</sup>。Gao等针对一类目标函数耦合和约束耦合的线性子系统提出了一种合作式分布式模型预测控制(CDMPC)算法<sup>[67]</sup>。

在国内,董海等将分布式模型预测控制方法应用于网络环境下的供应链管理,从而使得市场的动态变化得到快速响应的同时,有效解决了供应链管理中的动态优化控制问题<sup>[68]</sup>。张永等采用分布式模型预测控制来控制自动发电控制(AGC)系统中,使得AGC系统的调节速度与精度之间的协调性得到有效提高<sup>[69]</sup>。陈庆等针对子系统间呈现为串联结构关系的生产过程,提出了一种基于网络通信环境下的分布式预测控制方法<sup>[70]</sup>。张舒展等针对常减压塔的串级系统,依据其复杂的串联特性,结合工业环境下网络结构的特点,提出了一种基于纳什最优的分布式预测控制算法<sup>[71]</sup>。李纹等针对切换序列事先确定的一类非线性切换系统,设计了一种带可测延时和通信信道扰动的分布式预测控制控制器<sup>[72]</sup>。

## 1.5 论文的主要工作及内容安排

本文主要研究了传统动态矩阵控制方法的改进及其在分数阶系统中的应用。主要工作可以大致分为三个部分,第一部分为第三章,主要介绍了两种改进的动态矩阵控制方法,并在稳定汽油蒸汽压力控制系统中将二者与传统的动态矩阵控制进行对比分析,仿真结果表明了两种改进 DMC 方法在整体性能上的优势。第二部分为第四章,将第三章中两种改进 DMC 方法中控制性能最为突出的 PID 型动态矩阵控制方法运用于一类单入单出分数阶线性系统中,并在加热炉温度控制系统的仿真实例中验证了该方法的控制性能。第三部分为第五章,主要是考虑了一类多变量分数阶系统,通过将 PID 型动态矩阵控制方法推广至分布式控制结构中,得到一种改进的分布式动态矩阵控制方法,并将其用于控制一类多变量的分数阶系统,最后通过 MATLAB 仿真研究验证了所提方法的可行性和有效性。本文的具体内容安排如下:

第一章主要从分数阶系统控制理论、预测控制以及分布式预测控制三个方面阐述了本文研究课题的发展主干及研究现状。

第二章主要介绍了分数阶微积分的相关知识,并对动态矩阵控制、分布式模型预测控制的理论基础做了简要概述。

第三章在传统动态矩阵控制的基础上讨论了两改进的动态矩阵控制算法,并通过一个稳定汽油蒸汽压力控制系统的实例对三种 DMC 算法的控制效果进行对比分析,仿真结果表明了改进 DMC 方法的可行性。

第四章将改进的 DMC 算法运用于一类单入单出分数阶线性系统中, 并通过电加热炉温度控制系统的仿真实例, 对该算法设计的分数阶控制器和常规整数阶控制器的控制性能进行对比, 仿真结果验证了所提方法的有效性。

第五章通过在分布式的控制结构引入改进的 DMC 算法, 得到一种改进的分布式动态矩阵控制, 并将其应用于一类分数阶的多变量控制系统, 最后通过仿真研究对所提方法的控制性能进行说明。

第六章是对全文内容及未来研究方向的概括, 在对论文工作做出简要回顾的同时, 也对论文所研究领域的进一步发展做出了展望。



## 第2章 理论基础

### 2.1 分数阶微积分

分数阶微积分，也可称作广义微积分学，是对传统微积分学进行推广而来，即将微分方程中对函数的导数阶次选为分数，而不再只是整数。其实质上是非整数阶的微积分或任意阶次的微积分，阶次数可以是实数甚至复数。分数阶微积分通常用  ${}_0 D_t^\alpha f(t)$  来表示，微积分运算的上限和下限分别用  $t_0$  和  $t$  表示，微积分的阶次数用  $\alpha$  来表示（ $\alpha < 0$  时表示积分运算， $\alpha > 0$  时则表示微分运算），它可以为任意复数，本文将范围限定在实数域。

#### 2.1.1 分数阶微积分的定义与性质

分数阶微积分到目前为止尚不存在一个统一的定义，这也是其还未得到广泛应用的原因之一。随着分数阶微积分理论不断发展，早在两个世纪前分数阶微积分的许多定义和研究就已提出，所提的这些分数阶微积分定义<sup>[73]</sup>主要包括，如分数阶微积分的Grünwald-Letnikov(G-L)定义，Riemann-Liouville(R-L)定义以及Caputo定义等，研究者们分别站在不同的立场对几种分数阶微积分定义进行诠释，其可行性在实际工程中得以说明。下面将分别介绍这三种分数阶微积分定义。

##### (1) Grünwald-Letnikov分数阶微积分定义

分数阶微积分的G-L定义在分数阶控制中的应用较为常见，利用该定义可直接推导出分数阶导数的数值算法，具体描述形式如下<sup>[74]</sup>：

$$\begin{aligned} {}_0^{G-L} D_t^\alpha f(t) &= \lim_{h \rightarrow 0} h^{-\alpha} \sum_{j=0}^{[(t-t_0)/h]} (-1)^j \binom{\alpha}{j} f(t-jh) \\ &= h^{-\alpha} \sum_{j=0}^{[(t-t_0)/h]} \omega_j^{(\alpha)} f(t-jh) \end{aligned} \quad (2.1a)$$

$$\text{其中, } \binom{\alpha}{j} = \frac{\alpha(\alpha-1)(\alpha-2)\cdots(\alpha-j+1)}{j!} = \frac{\alpha!}{j!(\alpha-j)!} \quad (2.1b)$$

$t_0$  表示初始时间， $h$  表示采样步长， $[(t-t_0)/h]$  表示  $(t-t_0)/h$  的整数部分，

$\omega_j^{(\alpha)} = (-1)^j \binom{\alpha}{j}$  表示多项式系数，该系数可由下面的递推公式直接求得：

$$\omega_0^{(\alpha)} = 1, \omega_j^{(\alpha)} = \left(1 - \frac{\alpha+1}{j}\right) \omega_{j-1}^{(\alpha)}, \quad j=1, 2, \dots \quad (2.1c)$$

假设区间  $[t_0, t]$  上的函数  $f(x)$  存在  $m+1$  阶连续导数，且满足  $m = [\alpha]$   $m < \alpha < m+1$ ，对式(2.1a)中的定义式取极限可以得到：

$$\begin{aligned}
 {}^{G-L}D_t^\alpha f(t) &= \lim_{\substack{h \rightarrow 0 \\ nh=t-t_0}} \frac{1}{h^\alpha} \sum_{j=0}^n (-1)^j \binom{\alpha}{j} f(t-jh) \\
 &= \sum_{i=0}^m \frac{f^{(i)}(t_0)(t-t_0)^{-\alpha+i}}{\Gamma(-\alpha+i-1)} + \frac{1}{\Gamma(i-\alpha+1)} \int_{t_0}^t (t-\xi)^{i-\alpha} f^{(i)}(\xi) d\xi
 \end{aligned} \quad (2.2)$$

其中,  $\Gamma(z)$  为Gamma函数 (即Euler第二积分), 是推广到实数域及复数域上的阶乘函数, 可以通过公式  $\Gamma(z) = \int_0^\infty e^{-t} t^{z-1} dt$  来计算其数值。

假设函数  $f(t)$  满足零初始条件即  $f^{(i)}(t_0) = 0$ , 则对于任意实数  $p, q$ , 微分算子有如下性质:

$${}_{t_0}D_t^q({}_{t_0}D_t^p f(t)) = {}_{t_0}D_t^p({}_{t_0}D_t^q f(t)) = {}_{t_0}D_t^{p+q} f(t) \quad (2.3)$$

## (2) Riemann-Liouville分数阶微积分定义

分数阶微积分的R-L定义作为G-L定义的一种改进形式在当前的应用最为广泛, 其分数阶积分形式可描述为<sup>[74]</sup>:

$${}^{R-L}D_t^\alpha f(t) = \frac{1}{\Gamma(-\alpha)} \int_{t_0}^t (t-\tau)^{-\alpha-1} f(\tau) d\tau, \quad t > 0, \alpha \in \mathbb{R}^+ \quad (2.4)$$

R-L定义的分数阶微分形式是通过首先对函数  $f(t)$  做  $n-\alpha$  次积分运算, 然后再做  $n$  次微分运算所得, 其表达式描述如下<sup>[74]</sup>:

$${}^{R-L}D_t^\alpha f(t) = \frac{1}{\Gamma(n-\alpha)} \frac{d^n}{dt^n} \int_{t_0}^t \frac{f(\tau)}{(t-\tau)^{\alpha-n-1}} d\tau, \quad 0 \leq n-1 < \alpha < n \quad (2.5)$$

将R-L定义的积分形式与微分形式相结合即可得到如下统一形式:

$${}^{R-L}D_t^\alpha f(t) = \frac{1}{\Gamma(n-\alpha)} \left( \frac{d}{dt} \right)^n \int_{t_0}^t \frac{f(\tau)}{(t-\tau)^{1-(n-\alpha)}} d\tau, \quad n-1 < \alpha < n, n \in \mathbb{N} \quad (2.6)$$

由R-L定义又可得到其Laplace变换:

$$L\{{}_{t_0}^{R-L}D_t^\alpha f(t)\} = s^\alpha F(s) - \sum_{k=0}^{m-1} s^k {}_{t_0}^{R-L}D_t^{\alpha-k-1} f(t) \big|_{t=0^+} \quad (2.7)$$

其中,  $F(s)$  表示  $f(t)$  的原函数。

在零初始条件下, 可以进一步简写为:

$$L\{{}_{t_0}^{R-L}D_t^\alpha f(t)\} = s^\alpha F(s) \quad (2.8a)$$

由上式可以将  $\alpha$  阶的分数阶微积分算子写成如下传递函数形式:

$$H(s) = \frac{1}{s^\alpha} \quad (2.8b)$$

## (3) Caputo分数阶微积分定义

在处理工程中的实际问题时, 应该尽量防止物理意义不存在或者不明确的概念出现。而这样的问题正是R-L定义中所存在的, 如在式(2.6)中的初值定义虽然存在物理意义, 但却无法给出明确的物理解释<sup>[75]</sup>。Caputo针对这样一类问题, 提出了初值为整数阶导数的Caputo微积分定义, 其积分形式可描述如下<sup>[74]</sup>:

$${}^C D_t^{-\alpha} f(t) = \frac{1}{\Gamma(\alpha)} \int_{t_0}^t (t-\tau)^{\alpha-1} f(\tau) d\tau, \quad 0 < \alpha < 1 \quad (2.9)$$

其中,  $\Gamma(\cdot)$  为Gamma函数。

Caputo定义的微分形式可描述为:

$${}_t^C D_t^\alpha f(t) = \frac{1}{\Gamma(\alpha - n)} \int_{t_0}^t \frac{f^{(n)}(\tau)}{(t - \tau)^{\alpha - n + 1}} d\tau, \quad n - 1 < \alpha < n \quad (2.10)$$

由Caputo定义得到其Laplace变换:

$$L\{{}_t^C D_t^\alpha f(t)\} = s^\alpha F(s) - \sum_{k=0}^{m-1} s^{\alpha - k - 1} \left[ \frac{d^k f(t)}{dt^k} \right]_{t=0^+} \quad (2.11)$$

#### (4) 三种定义间的关系与性质

在分数阶微积分理论中, 其R-L定义与Caputo定义均是对G-L定义的改进和推广。当函数  $f(t)$  存在  $m+1$  阶的连续导数并且  $m$  至少可取  $[\alpha] = n-1$  时, 那么可认为 G-L 定义和 R-L 定义完全等价。若函数  $f(t)$  满足条件  $f^{(k)}(C) = 0, k = 0, 1, \dots, n-1, C$  为常数, 那么当  $f(t)$  有  $m+1$  阶的连续导数, 且至少可取得  $[\alpha] = n-1$  时, Caputo定义与G-L定义才相互等价。这里用  $N$  来代表自然数, 分别针对R-L定义以及Caputo定义的微分形式求取其在  $\alpha \rightarrow n$  下的极限, 即可推出  ${}_t^C D_t^n f(t) = {}_t^{G-L} D_t^n f(t) = s^{(n)}(t), \alpha = n \in N$ 。这就说明Caputo定义和R-L定义的积分形式等价。当  $\alpha$  的取值为正整数或负实数时, Caputo定义和R-L定义相互等价, 且存在如下的关系式:

$${}_t^{R-L} D_t^n f(t) = \sum_{k=0}^{n-1} \frac{f^{(k)}(t_0)(t - t_0)^{k-t_0}}{\Gamma(k - p + 1)} + {}_t^C D_t^n f(t) \quad (2.12)$$

Caputo定义与R-L定义的不同之处主要表现为对常数的求导, 对常数的导数有界且值为0的分数阶微积分定义为Caputo定义, 而对常数的导数无界的分数阶微积分定义则为R-L定义。综上所述, R-L定义可以简化分数阶导数中的运算, 因而更适合用在理论分析中; Caputo定义的引进则可以简化Laplace变换形式。

设函数  $f(t)$ 、 $g(t)$  均存在分数阶微积分,  $a$  表示任意常数,  $\alpha$ 、 $\beta$  表示任意数, 则分数阶微积分中的基本性质主要包括:

##### 1) 线性性质

$${}_t^C D_t^\alpha [af(t) + bg(t)] = a {}_t^C D_t^\alpha f(t) + b {}_t^C D_t^\alpha g(t) \quad (2.13)$$

##### 2) 交换性质和结合性质

$${}_t^C D_t^\alpha \left[ {}_t^C D_t^\beta f(t) \right] = {}_t^C D_t^{(\alpha+\beta)} f(t) = {}_t^C D_t^\beta \left[ {}_t^C D_t^\alpha f(t) \right] \quad (2.14)$$

##### 3) 如果 $f(t)$ 解析, 则 ${}_t^C D_t^\alpha f(t)$ 对于 $t$ 和 $\alpha$ 同样解析。

##### 4) 记忆性质

$f(t)$  的分数阶微积分的求解和所有在  $t$  时刻前的函数值均相关, 这个特性被称为分数阶微积分的记忆性质。

#### 2.1.2 分数阶系统的描述

由于整数阶微分方程很难精确描述现实世界中的很多物理及化学特性, 而

分数阶次的微分方程恰能对系统中被整数阶微分方程所忽略的特性加以考虑，使得其对于系统内在特性的描述更加全面。

典型的单入单出 (SISO) 分数阶线性系统的微分方程可用如下形式来表示：

$$\begin{aligned} y(t) + a_1 D^{\alpha_1} y(t) + a_2 D^{\alpha_2} y(t) + \cdots + a_{m_1} D^{\alpha_{m_1}} y(t) \\ = b_0 D^{\beta_0} u(t) + b_1 D^{\beta_1} u(t) + \cdots + b_{m_2} D^{\beta_{m_2}} u(t) \end{aligned} \quad (2.15)$$

其中，任意实数  $a_i (i=1, 2, \dots, m_1), b_j (j=0, 1, \dots, m_2)$  用来表示分数阶的阶次数，且满足  $\alpha_1 < \alpha_2 < \cdots < \alpha_{m_1}, \beta_0 < \beta_1 < \cdots < \beta_{m_2}$ ， $y(t)$ 、 $u(t)$  分别表示控制系统的输出量和输入量。

结合RL定义及其在(2.8a)中的Laplace变换，零初始条件下的分数阶系统可以用如下传递函数形式来描述：

$$G(s) = \frac{b_0 s^{\beta_0} + b_1 s^{\beta_1} + \cdots + b_{m_2} s^{\beta_{m_2}}}{1 + a_1 s^{\alpha_1} + a_2 s^{\alpha_2} + \cdots + a_{m_1} s^{\alpha_{m_1}}} \quad (2.16)$$

如果  $\alpha_i = i\alpha (i=1, 2, \dots, n_1)$ ， $\beta_j = j\beta (j=1, 2, \dots, n_2)$ ，则将该系统称为“同元次”的分数阶系统，因此式(2.16)又可写成如下形式：

$$G(s) = \frac{\sum_{j=1}^{n_2} b_j s^{j\beta}}{\sum_{i=1}^{n_1} a_i s^{i\alpha}} \quad (2.17)$$

## 2.2 动态矩阵控制

模型预测控制（又可简称为预测控制）作为先进控制中的一类典型计算机控制算法，对工业过程中复杂对象的控制具有极其重要的意义。一方面，预测控制可以用于控制大时滞环境中的不确定对象。另外一方面预测控制所要求的模型精度往往较低，因而在炼油及化工等各行业得以广泛运用，同时取得的经济效益十分可观。常用的预测控制算法主要有 MPHC、MAC、DMC、GPC 等，其中动态矩阵控制(DMC)作为工业过程中运用最为广泛的预测控制算法，一直为诸多学者所青睐。下面将针对 DMC 算法的具体实现及参数设计进行具体介绍。

### 2.2.1 动态矩阵控制的算法实现

DMC 基于对象的阶跃响应数据设计算法，在每个采样时刻，将经过反馈校正后的模型预测输出值代入到当前时刻的目标函数中，通过求取该目标函数的极值即令其导数为零，即可求得当前采样时刻下对象输入控制增量序列的最优解。选取最优控制序列的首元素与前一时刻的最优控制律相加，即可得到当前时刻的最优控制律。该算法主要对于渐进稳定的线性对象较为适用<sup>[76]</sup>，而针对不稳定过程对象，一般先利用 PID 控制稳定该对象，然后对稳定的过程对象实

施动态矩阵控制。针对非线性系统的处理方式通常有两种：第一，先利用泰勒展开或对李雅普诺夫函数中平衡点的局部线性化等方法得到线性化处理后的系统，然后再对其实施动态矩阵控制。第二，利用 T-S 模糊模型或人工神经网络模型等对非线性系统进行描述，然后对所描述的系统模型实施动态矩阵控制。本节分别从预测控制的三大基本原理(即预测模型、滚动优化及反馈校正)入手，对动态矩阵控制的算法实现进行介绍。

### (1) 预测模型

首先给被控对象输入一个单位阶跃信号，并依据实际工况选取合理的采样步长  $T_s$ ，即可测得对象的阶跃响应采样值  $a_i = a(iT_s), i = 1, 2, \dots$ 。对渐进稳定的对象而言，其阶跃响应将在某一时刻  $t_N = NT$  后趋于平稳，当  $a_i (i > N)$  与  $a_N$  的误差已减小至与量化误差及测量误差的数量级相同时，即可认为  $a_N$  近似等于该对象的阶跃响应稳态值。因此可以通过有限的阶跃响应数据集合  $\{a_1, a_2, \dots, a_N\}$  对被控对象的动态信息进行近似描述，该集合的组成元素即为被控对象的模型参数，建立如下形式的模型向量  $a$ ：

$$a = [a_1, a_2, \dots, a_N]^T \quad (2.18)$$

利用式(2.18)中的模型向量  $a$ ，可以建立对象的动态矩阵，其形式如下：

$$A = \begin{pmatrix} a_1 & 0 & \cdots & 0 \\ a_2 & a_1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ a_P & a_{P-1} & \cdots & a_{P-M+1} \end{pmatrix} \quad (2.19)$$

其中， $A$  表示被控对象  $P \times M$  阶的动态矩阵， $P$  表示动态矩阵控制中的优化时域长度， $M$  表示动态矩阵控制中的控制时域长度，且有  $M < P < N$ 。

在  $k-1$  时刻给被控对象施加控制增量  $\Delta u(k-1)$  得到其模型预测值  $y_p(k-1)$ ：

$$y_p(k-1) = y_M(k-1) + A_0 \Delta u(k-1) \quad (2.20a)$$

其中，

$$y_p(k-1) = \begin{bmatrix} y_1(k|k-1) \\ y_1(k+1|k-1) \\ \vdots \\ y_1(k+N-1|k-1) \end{bmatrix}, y_M(k) = \begin{bmatrix} y_0(k|k-1) \\ y_0(k+1|k-1) \\ \vdots \\ y_0(k+N-1|k-1) \end{bmatrix}, A_0 = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_N \end{bmatrix} \quad (2.20b)$$

$y_1(k|k-1), y_1(k+1|k-1), \dots, y_1(k+N-1|k-1)$  分别表示加入控制增量  $\Delta u(k-1)$  后被控对象在  $k-1$  时刻对  $k, k+1, \dots, k+N-1$  时刻的模型预测值， $y_0(k|k-1), y_0(k+1|k-1), \dots, y_0(k+N-1|k-1)$  分别表示被控对象在  $k-1$  时刻对  $k, k+1, \dots, k+N-1$  时刻的预测初值， $A_0$  表示基于阶跃响应数据建立的矩阵， $\Delta u(k-1)$  表示  $k-1$  时刻施加的控制增量。

在  $k$  时刻，对被控对象施加  $M$  个连续的控制增量  $\Delta u(k), \dots, \Delta u(k+M-1)$  得

到其预测输出值  $y_{PM}(k)$  为:

$$y_{PM}(k) = y_{p0}(k) + A\Delta u_M(k) \quad (2.21a)$$

其中,

$$y_{PM}(k) = \begin{bmatrix} y_M(k+1|k) \\ y_M(k+2|k) \\ \vdots \\ y_M(k+P|k) \end{bmatrix}, y_{p0}(k) = \begin{bmatrix} y_0(k+1|k) \\ y_0(k+2|k) \\ \vdots \\ y_0(k+P|k) \end{bmatrix}, \Delta u_M(k) = \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+M-1) \end{bmatrix} \quad (2.21b)$$

$y_M(k+1|k), y_M(k+2|k), \dots, y_M(k+P|k)$  表示被控对象在  $k$  时刻对  $k+1, k+2, \dots, k+P$  时刻的模型预测输出值,  $y_0(k+1|k), y_0(k+2|k), \dots, y_0(k+N|k)$  表示被控对象在  $k$  时刻对  $k+1, k+2, \dots, k+P$  时刻的预测初值。

## (2) 滚动优化

通过求解动态矩阵控制性能指标在每一时刻的最优问题, 可以得到该时刻的最优控制序列  $\Delta u_M(k)$ , 然后选取其首项作用于过程对象, 直到下一个时刻, 通过更新该时刻的校正模型和性能指标来求解当前时刻的最优控制序列  $\Delta u_M(k+1)$ , 并重复上述过程。依据上述滚动优化思想, 最后可以求得整个过程的最优控制律。

选取动态矩阵控制在  $k$  时刻的参考轨迹及性能指标, 其形式如下:

$$\min J(k) = \| (ref(k) - y_{PM}(k)) \|_Q^2 + \| \Delta u(k) \|_r^2 \quad (2.22a)$$

$$= Q(ref(k) - y_{p0}(k) - A\Delta u(k))^2 + r\Delta u^2(k) \quad (2.22b)$$

$$ref(k) = [ref_1(k), ref_2(k), \dots, ref_p(k)] \quad (2.22b)$$

$$ref_i(k) = \beta^i y(k) + (1 - \beta^i)c(k)$$

$$Q = diag(q_1, q_2, \dots, q_p) \quad (2.22c)$$

$$R = diag(r_1, r_2, \dots, r_M)$$

其中,  $Q$  表示误差权矩阵,  $q_1, q_2, \dots, q_p$  表示误差权矩阵的权系数;  $\beta$  表示柔化系数,  $c(k)$  表示过程对象的期望设定值;  $R$  表示控制权矩阵,  $r_1, r_2, \dots, r_M$  表示控制权矩阵的权系数,  $ref(k)$  表示  $k$  时刻的过程对象参考轨迹,  $ref_i(k)$  表示其中第  $i$  个参考点的取值。

结合上述式子, 对  $k$  时刻的性能指标  $J(k)$  求极值, 即令  $dJ(k)/d\Delta u_M(k) = 0$ , 求得最优控制输入增量为:

$$\Delta u_M(k) = (A^T Q A + R)^{-1} A^T Q (w_p(k) - y_{p0}(k)) \quad (2.23)$$

对过程对象在线实施动态矩阵控制算法之前, 可以先离线计算出常向量  $(A^T Q A + R)^{-1} A^T Q$  的值, 然后利用工控机中的软件平台将其存储在计算机内存中。然而由于  $w_p(k) - y_{p0}(k)$  涉及到相关矩阵运算, 使得该算法存在过大的计算量。针对存在较大时滞的过程对象, 运用该控制算法较为合适, 而针对于快速随动系统的过程对象, 该方法就难以达到理想的控制效果了。

## (3) 反馈校正

由于模型失配及外界扰动等不确定性因素在实际生产过程中普遍存在，可能会使模型输出的预测值和实际值之间产生一定偏差。如果不及时对产生的偏差进行修正，那么下一时刻的优化控制将会建立在当前不准确的预测信息基础上，最终会使得整个生产过程受到严重影响。因此，有必要在利用实时信息对每一时刻的模型预测值进行反馈校正。

利用软测量或传感器等方式在  $k+1$  时刻测得对象的实际输出值  $y(k+1)$ ，并与该时刻的模型预测输出值  $y_1(k+1|k)$  对比构成如下的输出误差：

$$e(k+1) = y(k+1) - y_1(k+1|k) \quad (2.24)$$

利用式(2.24)中的输出误差，通过加权方式来修正未来预测的输出值，具体方法可描述为：

$$\hat{y}_{cor}(k+1) = y_p(k) + he(k+1) \quad (2.25a)$$

其中，

$$\hat{y}_{cor}(k+1) = [\hat{y}_{cor}(k+1|k+1), \hat{y}_{cor}(k+2|k+1), \dots, \hat{y}_{cor}(k+N|k+1)]^T \quad (2.25b)$$

$$h = [h_1, h_2, \dots, h_N]^T \quad (2.25c)$$

校正向量  $h$  的权系数分别为  $h_1, h_2, \dots, h_N$ 。

由于时间基点在  $k+1$  时刻会产生变化，未来预测的时间点将会移至  $k+2, k+3, \dots, k+N+1$ ，因此  $\hat{y}_{cor}(k)$  中的元素需要通过移位才能构成  $k+1$  时刻的初始预测输出：

$$y_{p0}(k+1) = S\hat{y}_{cor}(k+1) \quad (2.26a)$$

其中，

$$S = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & 1 \\ 0 & 0 & \dots & \dots & 1 \end{bmatrix} \quad (2.26b)$$

用来表示状态转移矩阵。

### 2.2.2 动态矩阵控制方法的参数设计

由于 DMC 中的设计参数一般会利用模型参数、控制参数及校正参数对受控系统控制性能产生间接影响，而这些参数本身与控制系统的快速性能、稳定性能、鲁棒性能及抗干扰性能等之间的关系并不存在直接的解析表达式，可以作为参数设计的定量参考指标。因此，动态矩阵控制方法中满足要求的参数一般是无法通过解析方法来唯一确定的。所以，动态矩阵控制的参数整定，通常是采用试凑与仿真相结合的方式来完成的。为便于参数的整定，下面将对参数的选择原则及其对受控系统性能的定性影响进行分析。

#### (1) 采样周期 $T_s$ 和模型长度 $N$

类似于一般采样控制中采样周期的选择原则，DMC 中的采样时间  $T_s$  必须依据奈奎斯特采样定理选取，与控制系统的动态响应信息及其类型有直接关系。依据被控物理量的类型，采样周期  $T_s$  的值通常可从毫秒级取到秒级。对于采样周期的选择，综合文献中给出的各种建议，总结如下：

对于单容的被控对象，可以取  $T_s \leq 0.1T_a$ ，其中  $T_a$  为对象的惯性时间常数；

对于振荡的被控对象，可以取  $T_s \leq 0.1T_e$ ，其中  $T_e$  为对象的振荡周期；

对于滞后的被控对象，可以取  $T_s \leq 0.25T_l$ ，其中  $T_l$  为对象的纯滞后时间。

虽然上述规则为 DMC 采样周期的选择提供了一定参考价值，但由于 DMC 控制算法基于非最小化模型设计，其采样周期选取与模型长度也是密切相关的。通常要求  $t = NT_s$  时刻后的阶跃响应采样值与稳态值近似相等，这样就可以利用模型参数  $a_i (i=1,2,\dots,N)$  对被控对象的动态信息进行完整描述。因此，减小  $T_s$  会增大对象的模型维数  $N$ 。如果  $T_s$  取值过小，不但会使得计算频率增大，同时模型维数在短时采样周期内的增大也会带来计算量的增大，从而对控制的实时性造成不利影响。所以从计算机的内存及控制的实时性出发，应该合理选取合适的采样时间  $T_s$ ，使得模型长度  $N$  的取值区间维持在 20~50。对于系统的抗干扰性而言，能够使得干扰产生的影响受到快速及时的抑制作用。

## (2) 优化时域 $P$ 和误差权矩阵 $Q$

优化时域  $P$  就是从  $k$  时刻起对象输出轨迹趋近期望参考轨迹需要经过的步数，而误差权矩阵  $Q$  的组成元素  $q_i$  则表示输出轨迹在不同时刻趋近于参考轨迹的权重大小。优化时域的长度必须超出对象的滞后环节或其动态信息的主要部分，并包含最小相位系统的反向环节。除此之外，由于滞后时间及非最小相位特性均在系统的动态信息中保持不变，对象的输出轨迹在这个时间段内是无法追踪到其期望轨迹的，所以可以将滞后环节或反向环节对应的权值取为 0。为保证控制系统的稳定，通常会使用如下条件来选取合适的  $P$  和  $q_i$ ：

$$\left( \sum_{i=1}^P a_i q_i \right) a_N > 0 \quad (2.27)$$

$P$  的取值大小很大程度上影响了系统的稳定性和快速性：随着  $P$  取值的增大，系统的稳定性增强，而快速性则会减弱。若  $P$  取的足够小，例如取值为 1 时，在预测模型足够准确的前提下可实现一步最小拍控制，但这种情况同样也存在很大的局限性：一是可能在采样点之间存在波纹；二是考虑模型失配及扰动情况下的系统鲁棒性非常差；三是对于含时滞或非最小相位特性的被控对象不适用，非最小相位特性甚至会引起不稳定控制。如果  $P$  取的过大，可使得系统稳定性得到保证，但动态响应速度将会十分缓慢。因此  $P$  的长度取值应该以工程中对控制要求的稳定性及快速性为准则。

对于构成误差权矩阵  $Q$  的权系数，可选取为



$$q_i = \begin{cases} 0 \cdots a_i \text{时滞或反向部分} \\ 1 \cdots \text{其它情况} \end{cases} \quad (2.28)$$

然后再根据实际过程的输出误差及输出波动情况，对  $q_i$  做出适当调整。当过程输出误差及波动都比较大时，可将相应的  $q_i$  值适当调大。

### (3) 控制时域 $M$

控制时域的长度在 DMC 的目标函数中表示需要优化的控制变量个数。由于优化主要基于对象在未来  $P$  步输出值和期望值的偏差来实现的，最多只有  $P$  个控制变量有效，所以就物理意义而言一般有  $M \leq P$ 。从系统性能的角度上说：系统跟踪性能随着  $M$  的增大而增强，而系统的稳定性及鲁棒性则会变差。从计算量的角度来说：由于矩阵  $(A^T Q A + R)$  的维数即为  $M$  的大小，因此  $M$  的减小有利于减小控制系数的计算量。因此  $M$  的选取必须在兼顾计算量的同时还得考虑到系统的性能要求。

### (4) 控制权矩阵 $R$

在控制权矩阵  $R = \text{diag}(r_1, r_2, \dots, r_m)$  中，其加权系数  $r_j$  通常会取同一值，记作  $r$ 。控制权矩阵作为加入到 DMC 性能指标中的一种软约束，其作用主要是对控制增量产生的剧烈变化加以抑制，但其加入与系统稳定性的改善并无直接关系。而系统稳定性的改善主要通过参数  $P$  和  $M$  来实现，因此已不再是参数  $r$  调整时所关注的重点。引进  $r$  的主要目的在于对控制量的剧烈变化进行限制。所以进行参数整定时，可先设置  $r=0$ ，如果所控制的系统稳定而控制量的变化过大，那么可将  $r$  略微调大。实际过程中只要将  $r$  选取为一个很小的值，即可使得控制量变化趋近平缓。

### (5) 误差校正向量 $h$

误差校正向量  $h = [h_1, h_2, \dots, h_N]$  是 DMC 中唯一可以直接调整的运算参数，其中校正系数  $h_i$  的选取独立于其他参数设计之外。该向量仅仅在过程中存在的不确定性及环境扰动使得输出的预测值与实际值产生偏差时才起作用，而对过程对象动态响应的影响并不明显。校正系数  $h_i$  的选取方法如下：

$$\begin{aligned} h_1 &= 1 \\ h_i &= \alpha \\ i &= 2, 3, \dots, N \\ 0 &< \alpha \leq 1 \end{aligned} \quad (2.29)$$

随着  $\alpha$  减小将导致系统对干扰的敏感度下降从而使得抗干扰性减弱，但控制系统的鲁棒性增强；随着  $\alpha$  增大，控制系统的鲁棒性减弱而抗干扰性能增强。因此，对  $\alpha$  值的选取应该以工业过程中的实际需求为依据。

### (6) 柔化系数 $\beta$

参考轨迹中的柔化系数主要是为使过程对象的输出跟踪曲线更加柔和所设

定的。一般情况下，取值范围为  $0 \leq \beta < 1$ 。增大  $\beta$  的取值会使得系统跟踪曲线更加柔和，同时其跟踪速度随之减慢。

## 2.3 分布式模型预测控制

分布式模型预测控制的主要思想就是基于分布式控制结构将一个复杂大规模系统的预测控制优化问题分散到各个小规模 subsystem 中分布实现，并借鉴多智能体系统理论将分解所得的每个 subsystem 视为一个智能体，通过网络环境下各智能体物质、能量及信息的交互完成整个大规模系统的优化任务。在分布式预测控制中，依据纳什优化的思想，假设各智能体在  $k$  时刻已通过自身的性能指标获取到各自的最优解，并将本轮最优解同上一轮进行对比，然后在智能体之间通过网络通信共享各自的本轮最优解及对比结果，直至各智能体两轮所求最优解的偏差在允许误差范围内，则表示系统在当前时刻处于纳什平衡，其所求得的最优解即为当前时刻系统的纳什最优解，至此整个优化过程完成。最后将当前时刻纳什最优解的即时控制律分别作用于各个智能体，并将时域向后滚动一个时刻，重复整个优化过程，整个大规模系统的优化问题便得以解决。

### 2.3.1 分布式预测控制算法

整个大规模系统的优化控制可以视为  $N$  个智能体各自控制作用的合成，则系统在  $k$  时刻经过  $P$  个时域长度优化的模型预测输出可描述为<sup>[77]</sup>：

$$Y_{PM}(k) = f(Y_{p0}(k), \Delta u_{1,M}(k), \Delta u_{2,M}(k), \dots, \Delta u_{N,M}(k)) \quad (2.30)$$

其中，

$$Y_{PM}(k) = [Y(k+1|k), Y(k+2|k), \dots, Y(k+P|k)]^T$$

$$\Delta u_{i,M}(k) = [\Delta u_i(k), \Delta u_i(k+1), \dots, \Delta u_i(k+M-1)]^T \quad (i=1, 2, \dots, N)$$

$P$ 、 $M$  分别表示预测控制优化时域和控制时域的长度， $f$  表示映射关系的描述函数，包含一些满足平滑条件的非线性函数，系统在  $k$  时刻的初始预测值用  $Y_{p0}(k) = [Y_0(k+1|k), Y_0(k+2|k), \dots, Y_0(k+P|k)]^T$  来表示。

大规模系统的变量约束关系可以描述如下：

$$\begin{cases} \Delta u_{i,\min} \leq \Delta u_i(k+j) \leq \Delta u_{i,\max} \\ u_{i,\min} \leq u_i(k+j) \leq u_{i,\max} \\ Y_{\min} \leq Y(k+j|k) \leq Y_{\max} \end{cases} \quad (2.31)$$

假定有

$$\Delta u_i(k+j) = \Delta u_i(k+M-1) \quad (M \leq j \leq P) \quad (2.32)$$

则可以选取如下形式的系统性能指标：

$$\min_{\Delta u_{1,M}(k), \dots, \Delta u_{N,M}(k)} J = \sum_{j=1}^P L[Y(k+j|k), \Delta u_{1,M}(k), \Delta u_{2,M}(k), \dots, \Delta u_{N,M}(k)] \quad (2.33)$$

其中，函数  $L$  用来表示控制量和输出量的非线性映射关系，且性能指标在该函数

满足一定平滑条件的情况下有最优解。大规模系统的整体优化即为依据式(2.30)中的预测模型,并结合式(2.33)中的性能指标及式(2.31)中的约束条件,在每个采样时刻求解系统的全局最优解。当所要控制的对象为高维系统时,在控制时域长度 $M$ 的作用下,所需求解的优化变量 $\Delta u_{1,M}(k), \Delta u_{2,M}(k), \dots, \Delta u_{N,M}(k)$ 的维数也会很高,因此诸如此类大规模系统的非线性优化问题的在线求解对计算机的处理速度及性能要求极高,而实际工业过程中的计算机往往难以达到要求。

为克服常规计算机在处理速度和性能上的局限性,一般将整个大规模系统分解为 $N$ 个智能体子系统中,并在各子系统中分布求解各自的预测控制优化问题,便可在实际过程中应用上述预测控制算法。假设式(2.33)中的目标函数可以按智能体个数进行局部求解,则第 $i$ 个智能体的目标函数为:

$$\min_{\Delta u_{i,M}} J_i = \sum_{j=1}^P L_i[y_i(k+j|k), \Delta u_{i,M}(k)] \quad (2.34)$$

依据式(2.30)可以得到其模型预测输出:

$$Y_{i,PM}(k) = f[Y_{i,P0}(k), \Delta u_{1,M}(k), \Delta u_{2,M}(k), \dots, \Delta u_{N,M}(k)](i=1, \dots, N) \quad (2.35)$$

其中,

$$Y_{i,PM}(k) = [y_i(k+1|k), y_i(k+2|k), \dots, y_i(k+P|k)]^T,$$

$$Y_{i,P0}(k) = [y_{i,0}(k+1|k), y_{i,0}(k+2|k), \dots, y_{i,0}(k+P|k)]^T$$

经过上述过程,大规模系统的目标函数被分解为各小规模子系统的目标函数,而各智能体的模型预测输出依旧与所有的控制增量相关。为更好的处理分布式控制中多个性能指标的求解问题,通常将借鉴纳什优化的思想与理论,即假定各智能体子系统已经获取到其他智能体的最优解,因此仅仅依据自身的输入变量求解各自的性能指标,具体可以描述为:

$$\min_{\Delta u_i(k)} J_i | \Delta u_{j,M}^*(k) (j=1, 2, \dots, N, j \neq i) \quad (2.36)$$

进一步可以得到式(2.36)的等价形式

$$Y_{i,PM}(k) = f_i[Y_{i,P0}(k), \Delta u_{1,M}^*(k), \dots, \Delta u_{i-1,M}^*(k), \Delta u_{i,M}(k), \Delta u_{i+1,M}^*(k), \dots, \Delta u_{N,M}^*(k)]$$

$$\begin{cases} \Delta u_{i,\min} \leq \Delta u_i(k+j) \leq \Delta u_{i,\max} \\ u_{i,\min} \leq u_i(k+j) \leq u_{i,\max} \\ y_{i,\min} \leq y_i(k+j|k) \leq y_{i,\max} \end{cases} \quad (2.37)$$

通过式(2.37)求得第 $i$ 个智能体纳什意义下的最优解满足以下条件<sup>[77]</sup>:

$$J_i(\Delta u_{1,M}^*(k), \dots, \Delta u_{i-1,M}^*(k), \Delta u_{i,M}^*(k), \Delta u_{i+1,M}^*(k), \dots, \Delta u_{N,M}^*(k)) \leq J_i(\Delta u_{1,M}^*(k), \dots, \Delta u_{i-1,M}^*(k), \Delta u_{i,M}(k), \Delta u_{i+1,M}^*(k), \dots, \Delta u_{N,M}^*(k)) \quad (2.38)$$

通过上述的分布式预测控制算法,将一个大规模系统的在线优化求解问题转化为各小规模子系统的分布式优化问题,从而使得问题的规模及复杂性大大降低。与此同时,各子系统之间通过网络通信实现信息共享,在一定程度上弥补了分散式预测控制由于缺乏信息交流而造成控制性能变差的不足,并且依然

可使系统整体的控制性能得以保证。因此分布式预测控制算法可有效控制一类含约束的非线性高维系统。综上所述，整个分布式预测控制的滚动优化过程为：

第一步：首先对当前时刻各智能体的最优控制序列进行初步预测，得到  $\Delta u_{i,M}^l(k) = [\Delta u_i^l(k), \Delta u_i^l(k+1), \dots, \Delta u_i^l(k+M-1)]^T (i=1, \dots, N)$ ，其中  $l=0$ ，并通过网络与其他子系统交换信息。

第二步：分别求解各智能子系统的预测控制问题，求得本轮的迭代最优解  $\Delta u_{i,M}^{l+1}(k) (i=1, \dots, N)$ 。

第三步：判断各智能体是否已经符合迭代求解的收敛条件，即对于允许误差范围  $\varepsilon_i (i=1, \dots, N)$ ，是否有  $\|\Delta u_{i,M}^{l+1}(k) - \Delta u_{i,M}^l(k)\| \leq \varepsilon_i (i=1, \dots, N)$  成立。若所有的子系统均满足该条件，则  $\Delta u_{i,M}^*(k) = \Delta u_{i,M}^{l+1}(k) (i=1, \dots, N)$ ，迭代过程至此结束，并转至第四步；否则令  $\Delta u_{i,M}^l(k) = \Delta u_{i,M}^{l+1}(k) (i=1, \dots, N), l=l+1$ ，再回到第二步。

第四步：计算各智能体子系统在  $k$  时刻的即时控制律  $\Delta u_i(k) = [1 \dots 0] \Delta u_{i,M}^*(k) (i=1, \dots, N)$ ，并将其作为该智能体的最优控制作用于每个子系统中。

第五步：将时域向后滚动一个时刻，再跳转至第一步，重复以上步骤，直到系统完成全局优化。

### 2.3.2 分布式动态矩阵控制

为方便对分布式预测控制算法进行进一步说明，本节将考虑无约束的线性系统，并选用动态矩阵控制算法作为各智能体的优化算法。

对于一个大规模的控制系统，其控制目标是在满足整体性能最优的前提下使得系统的输出轨迹逼近参考轨迹。因此整个系统的预测输出模型及性能指标可描述为如下形式<sup>[77]</sup>：

$$Y_{PM}(k) = Y_{P0}(k) + A \Delta u_M(k) \quad (2.39a)$$

$$\min J(k) = \|\omega(k) - Y_{PM}(k)\|_Q^2 + \|\Delta u_M(k)\|_R^2 \quad (2.39b)$$

其中， $\omega(k)$  表示系统的参考轨迹， $\Delta u_M(k) = [\Delta u_{1,M}(k), \Delta u_{2,M}(k), \dots, \Delta u_{N,M}(k)]^T$ ， $A$  表示动态矩阵。假设系统可分解，则依据分布式思想可将整个系统分解为：

$$y_{i,PM}(k) = y_{i,P0}(k) + A_{ii} \Delta u_{i,M}(k) + \sum_{j=1, j \neq i}^n A_{ij} \Delta u_{j,M}(k) (i=1, 2, \dots, N) \quad (2.40)$$

其中，式(2.40)右边第三项用来表示其他智能体的输入量对第  $i$  个智能体输出量的影响， $A_{ii}, A_{ij}$  分别用来表示智能体  $i$  和智能体  $j$  的输入对智能体  $i$  输出的阶跃响应数据建立的动态矩阵，

$$A_{ij} = \begin{bmatrix} a_{ij}(1) & 0 & \cdots & 0 \\ a_{ij}(2) & a_{ij}(1) & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ a_{ij}(M) & a_{ij}(M-1) & \cdots & a_{ij}(1) \\ \vdots & \vdots & & \vdots \\ a_{ij}(P) & a_{ij}(P-1) & \cdots & a_{ij}(P+M-1) \end{bmatrix}, A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1N} \\ A_{21} & A_{22} & \cdots & A_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ A_{N1} & A_{N2} & \cdots & A_{NN} \end{bmatrix}$$

$a_{ij}(k)(i, j=1, \dots, N; k=1, 2, \dots)$  表示阶跃响应序列。

假设式(2.39b)中的系统性能指标可以按照  $N$  个智能体进行分解, 则第  $i$  个智能体的性能指标可以描述为:

$$\min J_i = \|\omega_i(k) - y_{i,PM}(k)\|_{Q_i}^2 + \|\Delta u_{i,M}(k)\|_{R_i}^2 \quad (i=1, 2, \dots, N) \quad (2.41)$$

其中,  $\omega_i(k) = [\omega_i(k+1), \omega_i(k+1), \dots, \omega_i(k+p)]^T (i=1, \dots, N)$  表示第  $i$  个智能体给定期望输出的参考轨迹,  $Q_i$ 、 $R_i$  分别表示第  $i$  个智能体的误差权矩阵与控制权矩阵。

根据纳什最优思想, 结合式(2.37)可以求得第  $i$  个智能体的纳什最优解:

$$\Delta u_{i,M}^*(k) = D_{ii}[\omega_i(k) - y_{i,p0}(k) - \sum_{j=1, j \neq i}^n A_{ij} u_{j,M}^*(k)] \quad (i=1, 2, \dots, N) \quad (2.42a)$$

其中,  $D_{ii} = (A_{ii}^T Q_i A_{ii} + R_i)^{-1} A_{ii}^T Q_i$ ,

进一步可以得到第  $i$  个智能体在  $k$  时刻的即时控制律为:

$$\Delta u_i^*(k) = [1, 0, \dots, 0] \Delta u_{i,M}^*(k) \quad (i=1, 2, \dots, N) \quad (2.42b)$$

重复上述步骤, 可以得到第  $i$  个智能体在  $k$  时刻的下一轮迭代最优解:

$$\Delta u_{i,M}^{l+1}(k) = D_{ii}[\omega_i(k) - y_{i,p0}(k) - \sum_{j=1, j \neq i}^n A_{ij} \Delta u_{j,M}^l(k)] \quad (2.43)$$

其中,  $\omega(k) = [\omega_1(k), \omega_2(k), \dots, \omega_N(k)]^T$ ,  $y(k) = [y_{1,p0}(k), y_{2,p0}(k), \dots, y_{N,p0}(k)]^T$ 。

假设本文所提的分布式预测控制算法收敛, 那么整个系统在  $k$  时刻的最优控制律为<sup>[77]</sup>:

$$\Delta u_M^{l+1}(k) = D_1[\omega(k) - y_{p0}(k)] + D_0 u_M^l(k) \quad (2.44)$$

其中,

$$D_1 = \begin{bmatrix} D_{11} & 0 & \cdots & 0 \\ 0 & D_{22} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & D_{NN} \end{bmatrix}, D_0 = \begin{bmatrix} 0 & -D_{11}A_{12} & \cdots & -D_{11}A_{1N} \\ -D_{22}A_{21} & 0 & \cdots & -D_{22}A_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ -D_{NN}A_{N1} & -D_{NN}A_{N2} & \cdots & 0 \end{bmatrix}$$

通过分析迭代算法的收敛性, 不难得出以下收敛定理:

定理: 分布式动态矩阵控制算法在纳什最优意义下的收敛条件是  $|\rho(D_0)| < 1$ , 其中  $\rho(D_0)$  用来表示矩阵  $D_0$  的谱半径。

证明: 假设  $k$  时刻的  $\omega(k)$ ,  $y_{p0}(k)$  均是已知的, 式(2.44)的第一项  $D_1[\omega(k) - y_{p0}(k)]$  是个与迭代不相关的常数, 因此算法的收敛性主要取决于式中的第二项, 即与下式的收敛条件等价

$$\Delta u_M^{l+1}(k) = D_0 \Delta u_M^l(k) \quad (2.45)$$

由此不难得出本节分布式迭代算法的收敛条件：

$$|\rho(D_0)| < 1 \quad (2.46)$$

证毕。

通过以无约束的多变量线性系统为例，并采用动态矩阵控制作为子系统控制策略，可推导出各智能体通过分布式迭代所求最优解的解析形式，最后的结果在形式上等价于解耦设计，但其实质是完全不同的。解耦设计是通过离线计算预先设计好的控制律，在滚动优化时便不再考虑彼此的耦合关系，而分布式预测控制则在每一个时刻滚动优化时均需考虑其他智能体对本身的关联作用，并在优化求解的过程中加以考虑，通过各智能体之间的信息交换来获取纳什意义上的最优解，因此这种算法能很好的控制子系统之间存在耦合关系的大规模系统，其特有的分布式求解思想使得在线计算的规模大大降低。

## 2.4 本章小结

本章首先介绍了分数阶微积分所涉及的相关知识，然后从算法实现及参数设计两个方面详细介绍了动态矩阵控制方法的基本原理及理论基础，最后通过在分布式控制结构中引入动态矩阵控制方法，对分布式预测控制算法的具体实现进行介绍，为后面章节对动态矩阵控制方法的进一步推广与应用做铺垫。

## 第3章 改进的动态矩阵控制算法

### 3.1 引言

动态矩阵控制(Dynamic Matrix Control, DMC)作为预测控制中一类极具代表性的控制方法,在工程应用中最为广泛。DMC 基于对象的阶跃响应建模,将传统自校正技术中的单步预测推广至多步预测,并依据测得的实际系统反馈信息进行滚动优化,从而使模型参数变化对控制算法的影响得到有效抑制,同时对实际中的模型失配及未知扰动等不确定因素的适应能力极强。然而 DMC 算法同时存在一些不足,如在线计算规模复杂,控制参数设计的自由度较低,不能及时抑制因系统未知扰动作用而造成的影响等。因此在传统 DMC 的基础上研究其算法的改进,将进一步推广 DMC 在实际工程中更好的应用。本章分别通过引入分数阶积分算子及依据 PID 的典型控制结构重新建立常规 DMC 方法中的性能指标,不仅弥补了传统的 DMC 算法在实际工程应用中的不足,同时使得算法的控制参数调节更加灵活。最后在稳定塔压力控制的仿真实例中,将改进的 DMC 与传统 DMC 的控制效果进行对比分析,仿真结果表明 PID 算子改进的 DMC 最优,分数阶算子改进的 DMC 次之。

### 3.2 动态矩阵控制方法的改进

#### 3.2.1 动态矩阵控制中预测模型的建立

一般情况下,实际工业中的大多数过程对象可以用一阶惯性加纯滞后(FOPDT)模型加以描述:

$$G(s) = \frac{K}{Ts+1} e^{-\tau s} \quad (3.1)$$

其中,  $K$  为过程对象的稳态增益,  $T$  为一阶惯性环节的时间常数,  $\tau$  为过程对象的纯滞后。

首先选取合适的采样周期  $T_s$ , 并通过过程的阶跃响应实验获取被控对象的阶跃响应数据  $a_1, a_2, \dots, a_t, \dots$ , 直至对象的阶跃响应在某一时刻  $t_N = NT_s$  趋于平稳即当  $a_t (t > N)$  与  $a_N$  的误差和测量误差有相同的数量级时, 建立控制系统的模型向量  $\mathbf{a}$ :

$$\mathbf{a} = [a_1, a_2, \dots, a_N]^T \quad (3.2)$$

其中,  $T$  为矩阵的转置符号,  $N$  为建模时域,  $a_N$  近似等于阶跃响应的稳态值。

基于所得模型向量建立过程的动态矩阵:

$$A = \begin{bmatrix} a_1 & 0 & \cdots & 0 \\ a_2 & a_1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ a_M & a_{M-1} & \cdots & a_1 \\ \vdots & \vdots & \ddots & \vdots \\ a_P & a_{P-1} & \cdots & a_{P-M+1} \end{bmatrix} \quad (3.3)$$

其中， $P$ 、 $M$  分别为动态矩阵控制算法的优化时域和控制时域，且满足  $M < P < N$ 。

在  $k-1$  时刻给过程施加控制增量  $\Delta u(k-1)$ ，可以得到对象的模型预测输出：

$$y_{N1}(k-1) = y_{N0}(k-1) + \mathbf{a}\Delta u(k-1) \quad (3.4)$$

其中，

$$y_{N1}(k-1) = [y_1(k|k-1), y_1(k+1|k-1), \dots, y_1(k+N-1|k-1)]^T$$

$$y_{N0}(k-1) = [y_0(k|k-1), y_0(k+1|k-1), \dots, y_0(k+N-1|k-1)]^T$$

$y_{N1}(k-1)$  表示过程对象在  $k-1$  时刻的模型预测输出向量， $y_{N0}(k-1)$  表示过程对象在  $k-1$  时刻的初始预测输出向量， $k+i|k$  用来表示  $k$  时刻对  $k+i$  时刻的预测。

在通过检测得到过程对象在  $k$  时刻的实际输出  $y(k)$  后，为补偿实际过程中由模型失配、环境扰动等未知因素引起的预测输出与实际输出的偏差，这里采取加权方式利用输出误差信息对未来的输出预测值进行修正，具体修正方法如下：

$$y_{cor}(k) = y_{N1}(k-1) + h e(k) \quad (3.5)$$

其中，

$$y_{cor}(k) = [y_{cor}(k|k), y_{cor}(k+1|k), \dots, y_{cor}(k+N-1|k)]^T$$

$$h = [1, \alpha, \alpha, \dots, \alpha]$$

$$e(k) = y(k) - y_1(k|k-1)$$

$y_{cor}(k)$  表示过程对象在  $k$  时刻进行误差校正后的模型预测输出向量， $e(k)$  表示  $k$  时刻过程对象实际输出与预测输出之间的误差值， $h$  表示误差补偿的权向量， $\alpha$  表示误差校正的权系数；

由于时间基点在  $k$  时刻会发生变化，即预测的未来时间点将移动至  $k+1, k+2, \dots, k+N$ ，因此  $y_{cor}(k)$  中的元素必须经过移位操作才能构成过程对象在  $k$  时刻的初始预测输出向量，具体形式如下：

$$y_{N0}(k) = S y_{cor}(k) \quad (3.6)$$

其中， $N \times N$  的状态转移矩阵  $S$  定义为



$$S = \begin{bmatrix} 0 & 1 & 0 & \cdots & \cdots \\ 0 & 0 & 1 & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots & \cdots \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix}$$

这里由于过程对象是渐进稳定的，其阶跃响应在最后某个时刻将趋于平稳，因此式 (3.4) 中  $y_0(k+N|k)$  可以由  $y_{cor}(k+N-1|k)$  进行替换。

在得到  $k$  时刻过程对象的初始预测输出之后，为了让模型的输出预测值尽可能地接近给定期望输出，假设对过程对象施加  $M$  个连续的控制增量  $\Delta u(k), \Delta u(k+1), \dots, \Delta u(k+M-1)$ ，得到其预测模型的输出值  $y_{PM}$

$$y_{PM}(k) = y_{p0}(k) + A\Delta u_M(k) \quad (3.7)$$

其中，

$$\begin{aligned} y_{PM}(k) &= [y_M(k+1|k), y_M(k+2|k), \dots, y_M(k+P|k)]^T \\ y_{p0}(k) &= [y_0(k+1|k), y_0(k+2|k), \dots, y_0(k+P|k)]^T \\ \Delta u_M(k) &= [\Delta u(k), \Delta u(k+1), \dots, \Delta u(k+M-1)]^T \end{aligned}$$

$y_{PM}(k)$  表示受到控制增量  $\Delta u_M(k)$  作用的过程对象在  $k$  时刻的模型预测输出向量， $y_{p0}(k)$  为  $k$  时刻过程模型的初始预测输出向量  $y_{N0}(k)$  的前  $P$  项。

### 3.2.2 动态矩阵控制性能指标的选取

#### (1) 基于分数阶算子改进的性能指标

在动态矩阵控制方法的目标函数中引入分数阶的积分算子<sup>[78]</sup>，并选取合适的参考轨迹，其形式如下：

$$\begin{aligned} \tilde{J}_{FDMC} &= \gamma_1 I_{T_s}^{PT_s} [y_r(t) - y_M(t)]^2 + \gamma_2 I_{T_s}^{MT_s} \Delta u(t-1)^2 \\ &= \int_{T_s}^{PT_s} D^{1-\gamma_1} [y_r(t) - y_M(t)]^2 dt + \int_{T_s}^{MT_s} D^{1-\gamma_2} \Delta u(t-1)^2 dt \end{aligned} \quad (3.8a)$$

$$y_r(k+i) = \lambda^i y_p(k) + (1-\lambda^i)c(k) \quad (3.8b)$$

其中， $\gamma_1, \gamma_2$  为任意实数， ${}^\gamma I_{t_1}^{t_2} f(t)$  表示函数  $f(t)$  在  $[t_1, t_2]$  上的  $\gamma$  次积分， $D$  为微分符号。

选取合适的采样周期  $T_s$ ，依据 G-L 分数阶微积分定义，对上述目标函数进行离散化可得：

$$\tilde{J}_{FDMC} \cong [Y_r(k) - y_{PM}(k)]^T \Lambda(\gamma_1, T_s) [Y_r(k) - y_{PM}(k)] + \Delta u_M(k)^T \Lambda(\gamma_2, T_s) \Delta u_M(k) \quad (3.9)$$

其中，

$$\begin{aligned} Y_r(k) &= [y_r(k+1), y_r(k+2), \dots, y_r(k+P)]^T \\ \Lambda(\gamma_\varepsilon, T_s) &= T_s^{\gamma_\varepsilon} \text{diag}(w_{P-1}, w_{P-2}, \dots, w_1, w_0) \end{aligned}$$

$$w_q = \omega_q^{(\gamma_\varepsilon)} - \omega_{q-(P-1)}^{(\gamma_\varepsilon)}$$

$\omega_0^{(\gamma_\varepsilon)} = 1$ ，当  $\forall q > 0$  时，有  $\omega_q^{(\gamma_\varepsilon)} = (1 - \frac{1 + \gamma_\varepsilon}{q})\omega_{q-1}^{(\gamma_\varepsilon)}$ ，当  $q < 0$  时，有  $\omega_q^{(\gamma_\varepsilon)} = 0$ ， $\varepsilon = 1, 2$ 。

在上式中进一步引入误差权矩阵  $Q = \text{diag}(q_1, q_2, \dots, q_p)$  和控制权矩阵  $R = \text{diag}(r_1, r_2, \dots, r_p)$  可得到：

$$J_{FDMC} = [Y_r(k) - y_{PM}(k)]^T \Lambda(\gamma_1, T_s) Q [Y_r(k) - y_{PM}(k)] + \Delta U^T \Lambda(\gamma_2, T_s) R \Delta U \quad (3.10)$$

对上述目标函数求极值即令  $\frac{\partial J_{FDMC}}{\partial \Delta u_M(k)} = 0$ ，求解得到如下形式的最优控制律：

$$\Delta u_M(k) = (A^T \Lambda(\gamma_1, T_s) Q A + \Lambda(\gamma_2, T_s) R)^{-1} A^T \Lambda(\gamma_1, T_s) Q (Y_r(k) - y_{PM}(k)) \quad (3.11a)$$

$$\Delta u(k) = [1, 0, \dots, 0] \Delta u_M(k) \quad (3.11b)$$

$$u(k) = u(k-1) + \Delta u(k) \quad (3.11c)$$

在  $k+l$  ( $l = 1, 2, 3, \dots$ ) 时刻，重复上述过程并依次循环求解动态矩阵控制器的即时控制量  $u(k+l)$ ，再将其作用于过程对象。

## (2) 基于 PID 算子改进的性能指标

在动态矩阵控制方法中引入 PID 控制中的比例、积分、微分系数，依据 PID 控制结构重新建立其目标函数<sup>[79]</sup>，具体形式描述如下：

$$J = [Y_r(k) - y_{PM}(k)]^T K_i [Y_r(k) - y_{PM}(k)] + [\Delta Y_r(k) - \Delta y_{PM}(k)]^T K_p [\Delta Y_r(k) - \Delta y_{PM}(k)] + \quad (3.12a)$$

$$[\Delta^2 Y_r(k) - \Delta^2 y_{PM}(k)]^T K_d [\Delta^2 Y_r(k) - \Delta^2 y_{PM}(k)] + \Delta u_M(k)^T R \Delta u_M(k)$$

$$y_r(k+i) = \lambda^i y(k) + (1 - \lambda^i) c(k) \quad (3.12b)$$

其中，

$$Y_r(k) = [y_r(k+1), y_r(k+2), \dots, y_r(k+P)]^T$$

$$\Delta Y_r(k) = [\Delta y_r(k+1), \Delta y_r(k+2), \dots, \Delta y_r(k+P)]^T$$

$$\Delta y_{PM}(k) = [\Delta y_M(k+1|k), \Delta y_M(k+2|k), \dots, \Delta y_M(k+P|k)]^T$$

$$\Delta^2 Y_r(k) = [\Delta^2 y_r(k+1), \Delta^2 y_r(k+2), \dots, \Delta^2 y_r(k+P)]^T$$

$$\Delta^2 y_{PM}(k) = [\Delta^2 y_M(k+1|k), \Delta^2 y_M(k+2|k), \dots, \Delta^2 y_M(k+P|k)]^T$$

$$\Delta y_r(k+i) = y_r(k+i) - y_r(k+i-1) \quad (3.12c)$$

$$\Delta y_M(k+i|k) = y_M(k+i|k) - y_M(k+i-1|k) \quad (3.12d)$$

$$\Delta^2 y_r(k+i) = \Delta y_r(k+i) - \Delta y_r(k+i-1) \quad (3.12e)$$

$$\Delta^2 y_M(k+i|k) = \Delta y_M(k+i|k) - \Delta y_M(k+i-1|k) \quad (3.12f)$$

$y_r(k+i)$  为给定期望输出的参考轨迹， $y(k)$  为  $k$  时刻过程实测输出， $c(k)$  为  $k$  时刻的期望输出， $\lambda$  为参考轨迹柔化因子， $K_p = \text{diag}(k_p, \dots, k_p)$ ， $K_i = \text{diag}(k_i, \dots, k_i)$ ， $K_d = \text{diag}(k_d, \dots, k_d)$  分别为比例系数矩阵、积分系数矩阵、

微分系数矩阵；  $R = \text{diag}(r_1, r_2, \dots, r_p)$  为控制加权系数矩阵。

定义

$$E_0(k) = Y_r(k) - y_{PM}(k) = [e_0(k+1), e_0(k+2), \dots, e_0(k+P)]^T \quad (3.13a)$$

$$\Delta E_0(k) = \Delta Y_r(k) - \Delta y_{PM}(k) = [\Delta e_0(k+1), \Delta e_0(k+2), \dots, \Delta e_0(k+P)]^T \quad (3.13b)$$

$$\Delta^2 E_0(k) = \Delta^2 Y_r(k) - \Delta^2 y_{PM}(k) = [\Delta^2 e_0(k+1), \Delta^2 e_0(k+2), \dots, \Delta^2 e_0(k+P)]^T \quad (3.13c)$$

将(3.13a), (3.13b), (3.13c)代入(3.12a)，可以将目标函数转化为如下形式

$$J = E_0(k)^T K_i E_0(k) + \Delta E_0(k)^T K_p \Delta E_0(k) + \Delta^2 E_0(k)^T K_d \Delta^2 E_0(k) + \Delta u_M(k)^T R \Delta u_M(k) \quad (3.14)$$

结合(3.12c) (3.12d) (3.13b) (3.13c)，可以得到

$$\begin{aligned} \Delta e_0(k+i) &= \Delta y_r(k+i) - \Delta y_M(k+i|k) \\ &= y_r(k+i) - y_M(k+i|k) - (y_r(k+i-1) - y_M(k+i-1|k)) \\ &= e_0(k+i) - e_0(k+i-1) \end{aligned} \quad (3.15a)$$

同理可得

$$\Delta^2 e_0(k+i) = \Delta e_0(k+i) - \Delta e_0(k+i-1) \quad (3.15b)$$

为方便起见，特别定义  $e_0(k) = \Delta e_0(k) = 0$

并引入一个矩阵

$$S_1 = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 \\ -1 & 1 & 0 & \dots & 0 \\ 0 & -1 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & -1 & 1 \end{bmatrix}_{P \times P} \quad (3.16)$$

进而有

$$\begin{cases} \Delta E_0(k) = S_1 E_0(k) \\ \Delta^2 E_0(k) = S_1 \Delta E_0(k) = S_1^2 E_0(k) \end{cases} \quad (3.17)$$

将(3.17)代入(3.14)，将目标函数进一步转化为：

$$\begin{aligned} J &= E_0(k)^T K_i E_0(k) + E_0(k)^T S_1^T K_p S_1 E_0(k) \\ &\quad + E_0(k)^T (S_1^2)^T K_d (S_1^2) E_0(k) + \Delta u_M(k)^T R \Delta u_M(k) \\ &= E_0(k)^T Q E_0(k) + \Delta u_M(k)^T R \Delta u_M(k) \end{aligned} \quad (3.18)$$

其中，  $Q = K_i + K_p S_1^T S_1 + K_d (S_1^2)^T (S_1^2)$

以  $\Delta u_M(k)$  为控制变量，通过求解目标函数(3.18)的最小值，由  $\frac{\partial J}{\partial \Delta u_M(k)} = 0$ ，

得到如下形式的最优控制律：

$$\Delta u_M(k) = (R + A^T Q A)^{-1} A^T Q (Y_r(k) - y_{p0}(k)) \quad (3.19)$$

取  $D^T = [1, 0, \dots, 0]_{1 \times P}$ ，可以得到被控对象所需实际控制增量  $\Delta u(k)$ ：

$$\Delta u(k) = D^T (R + A^T Q A)^{-1} A^T Q (Y_r(k) - y_{p0}(k)) \quad (3.20)$$

### 3.3 案例研究：稳定汽油蒸汽压力控制系统

### 3.3.1 稳定系统的工艺流程

在催化裂化过程中，吸收塔、解吸塔、稳定塔、再吸收塔以及其他相关的辅助装置共同组成了吸收稳定系统<sup>[80]</sup>。图 3.1 所显示的正是该系统的具体工艺流程图。该稳定系统的主要目的就是要将由分馏塔的塔顶分离出来的粗汽油和富气通过催化裂化作用进一步得到所需的干气、液化石油气以及满足要求的稳定汽油。

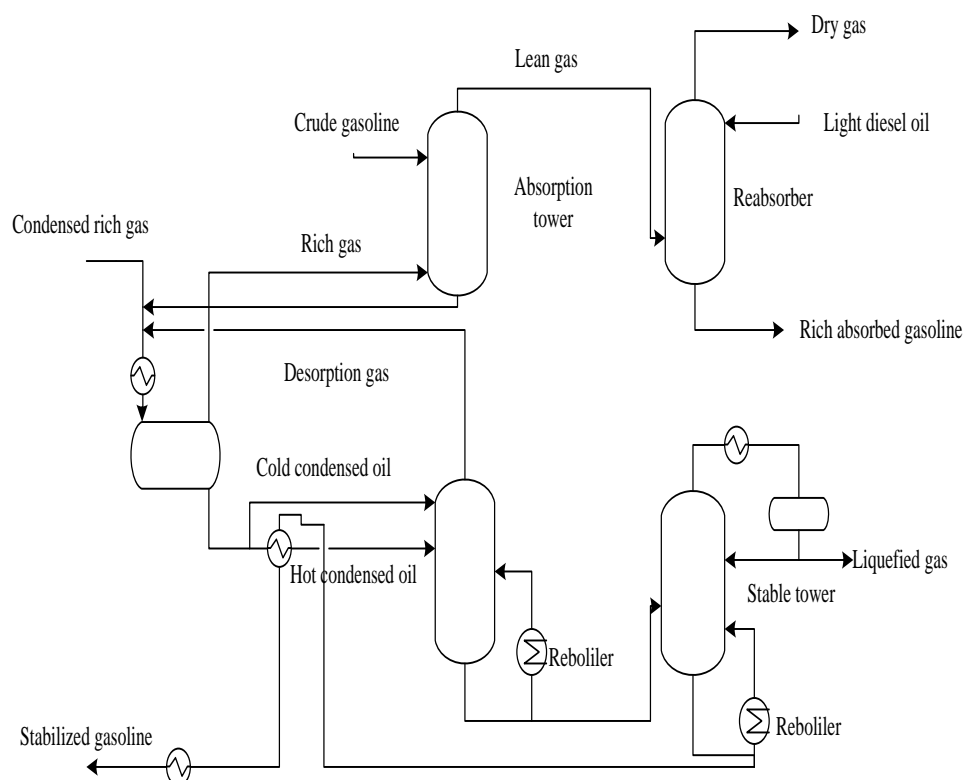


图 3.1 催化裂化过程中的吸收稳定系统

首先通过热传导将流入的压缩富气、富吸收汽油及解吸收气进行混合处理，并在平衡罐中平衡闪蒸<sup>[80]</sup>。平衡闪蒸后的富气经过平衡罐顶部流入吸收塔，富气中的轻烃成份通过吸收塔中吸收的粗汽油进行回收。吸收塔在经过回收处理剩下的气体称作贫气；贫气经过吸收塔的塔顶流入再吸收塔，并利用再吸收塔中的吸收剂做进一步分离处理后，在塔顶所得的产品称为干气。富吸收汽油由再吸收塔塔顶流入主分馏塔。由平衡罐流出的凝缩汽油可分为两类：一类经过与稳定汽油热交换后作为热进料流入解吸塔的中上部。另一类则作为冷进料有从解吸塔塔顶直接流入解吸塔，经过解吸塔塔顶分离得到解吸气。在解吸塔塔顶所得的脱乙烷汽油将流入稳定塔中做进一步分离，并进行加热处理，稳定塔在对脱乙烷汽油实施加热处理后，在塔顶得到液化石油气，在塔底得到的为稳定汽油<sup>[80]</sup>。稳定汽油主要流向也可分为两类：一类将从稳定塔流出，而另外一

类则通过与热凝缩油进行热交换后流入吸收塔并与最初的吸收剂相混合，并通过循环重复使用。

### 3.3.2 稳定系统的控制目标

在催化裂化的工业过程中，稳定汽油的质量和纯度的提高均是过程控制所追求的重要目标。稳定汽油的蒸汽压力作为催化裂化装置中的一个关键性指标，其大小在很大程度上决定了系统产生的稳定汽油质量和纯度。蒸汽压力过高或者过低均会对汽油的质量好坏产生影响。因此在催化裂化过程中，为使得产品质量与生产效率得到进一步提升，必须严格控制稳定汽油的蒸汽压力大小，设计满足实际工业过程需求的控制器，以加强控制系统的抗干扰能力及跟踪性能。

### 3.3.3 改进的 DMC 方法与传统 DMC 方法的比较

对控制系统输入一个单位阶跃信号，得到蒸汽压力对象的一系列阶跃响应数据。利用所得的阶跃响应数据，基于最小二乘法拟合得到系统的阶跃响应曲线。通过两点法计算得到 FOPDT 的模型参数：

$$G(s) = \frac{1.4}{9s+1} e^{-4s} \quad (3.21)$$

用  $y(k)$  表示稳定汽油的蒸汽压力， $u(k)$  表示蒸汽阀的开度，其中  $k$  为离散后的时刻。图 3.2 中显示了模型匹配情况下系统在基于分数阶算子改进的 DMC（FO-DMC）方法控制下的输出响应曲线随分数阶积分阶次参数变化的对比曲线。从图 3.2 中可以看出，系统的跟踪速度随着性能指标(3.10a)中的参数  $\gamma_1$  的增大而加快，随参数  $\gamma_2$  的增大而减慢。通过多次仿真实验可以发现，系统在  $\gamma_1 = 8$ ， $\gamma_2 = 1$  时的闭环响应性能最好。

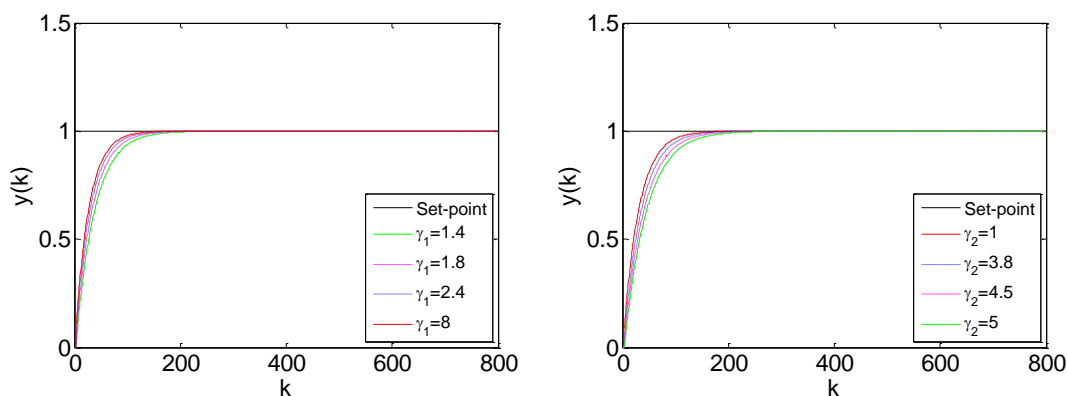


图 3.2 模型匹配时 FO-DMC 方法控制下系统随可调参数变化的闭环输出响应对比曲线

图 3.3 显示了模型匹配情况下基于 PID 算子改进的 DMC（PID-DMC）方法控制下的系统随 PID 参数变化的输出响应对比曲线。从图 3.3 中可以发现，系统的跟踪速度随着性能指标(3.11a)中比例系数  $k_p$  和微分系数  $k_d$  的增大而加快，随着积分系数  $k_i$  的增大而减慢。通过多次仿真实验发现，系统在  $k_p = 0.1$ ， $k_i = 0.1$ ， $k_d = 5.6$  时的闭环响应性能最好。

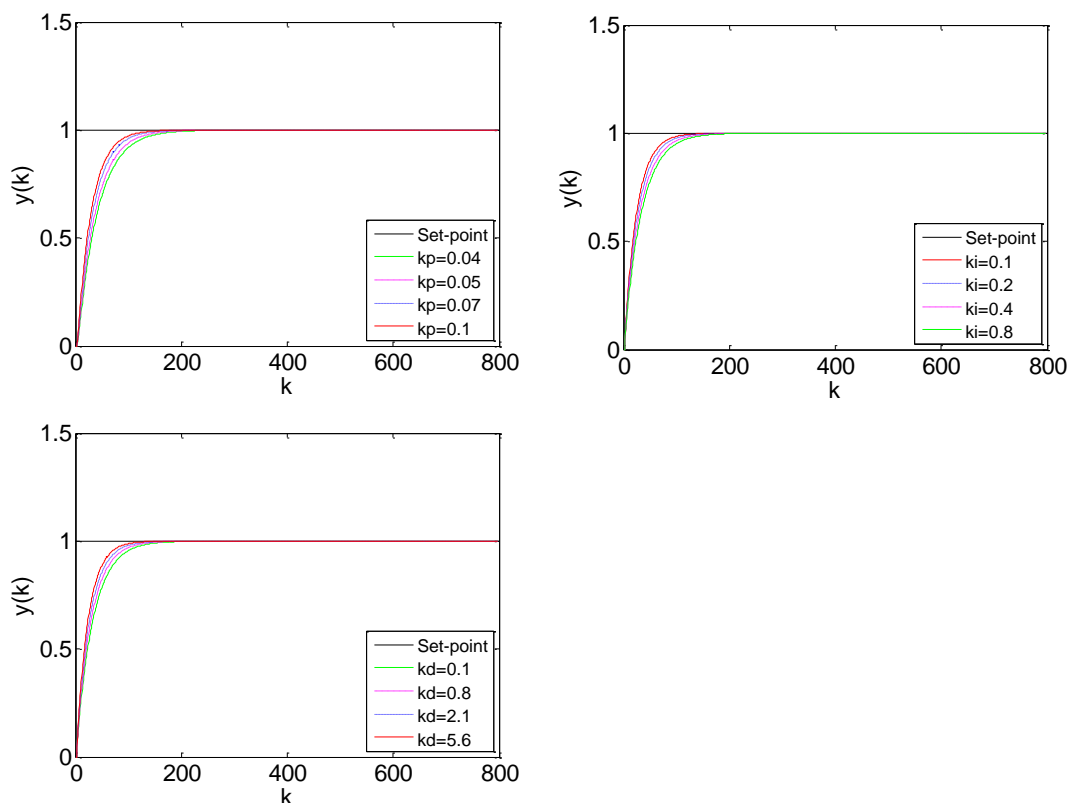


图 3.3 模型匹配时 PID-DMC 方法控制下系统随可调参数变化的闭环输出响应对比曲线

为方便进一步将两种改进的 DMC 算法同传统 DMC 算法进行对比, 设置如下的仿真参数:  $k=0$  时刻的设定值选取为 1; 在  $k=400$  的时刻, 给系统施加一个振幅为-0.1 的输出干扰; 表 3.1 中列出了本章中 PID-DMC 方法、FO-DMC 方法及传统 DMC 方法的参数详情:

表 3.1 两种改进的 DMC 方法及传统 DMC 方法的参数表

Parameters	PID-DMC	FO-DMC	DMC
$P$	14		
$M$	2		
$N$	60		
$T_s$	2s		
$\alpha$	0.6		
$\beta$	0.96		
$q_i$	\	1	1
$r_i$	0.05		
$\gamma_1$	\	8	\

$\gamma_2$	\	1	\
$k_p$	0.1	\	\
$k_i$	0.1	\	\
$k_d$	5.6	\	\

图 3.4 显示了 PID-DMC 方法、FO-DMC 方法与常规 DMC 方法控制下系统闭环响应对比曲线。从图 3.4 中可以看出，三种方法控制下的系统均无超调和振荡，两种改进的 DMC 方法在一定程度上改进了控制系统的跟踪性能和抗干扰能力，其中 PID-DMC 方法控制下的系统响应速度最快，且抗输出干扰的恢复能力最强，FO-DMC 方法下的响应速度和抗输出干扰能力则次之，而常规 DMC 方法是三者中最差的，因此在三种控制方法中，PID-DMC 方法控制的系统整体控制性能是最好的。

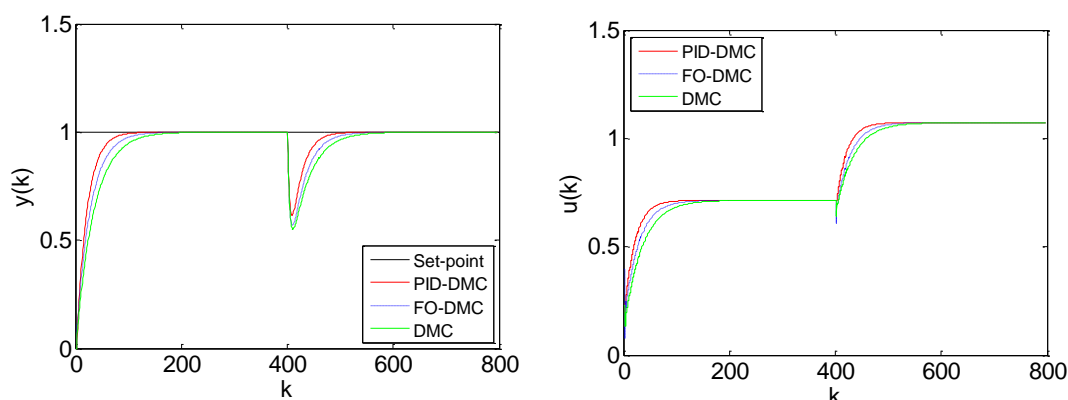


图 3.4 模型匹配时 PID-DMC、FO-DMC、DMC 方法控制下的系统闭环响应对比曲线

由于实际建模过程存在的各种不确定性因素，往往会出现建立的模型与实际过程对象不匹配的情况，即所谓的模型失配。本章将过程参数的最大失配程度假定为 $\pm 20\%$ ，利用蒙特卡洛方法随机产生三组过程失配参数：

第一组失配情况下的过程参数： $K=1.3474, T=8.6425, \tau=4.1236$ ；

第二组失配情况下的过程参数： $K=1.5013, T=7.8845, \tau=3.8415$ ；

第三组失配情况下的过程参数： $K=1.6352, T=10.1014, \tau=4.1427$ 。

在三组模型失配情况下，控制器的设置参数将与匹配情况下保持一致，此处不再说明。图 3.5-图 3.7 分别显示了两种改进的 DMC 方法与常规 DMC 方法在三组失配情况下的比较结果。从图中可以发现，考虑失配情况下的系统在三组控制方法下均能获得无超调和振荡的闭环响应，其中 PID-DMC 方法控制下的系统响应速度最快，且在系统受到输出干扰时的恢复能力最强，FO-DMC 方法则次之，而常规 DMC 方法仍然是三者中最差的。因此对比之下 PID-DMC 控制方法在三种方法控制下的系统整体性能最优。

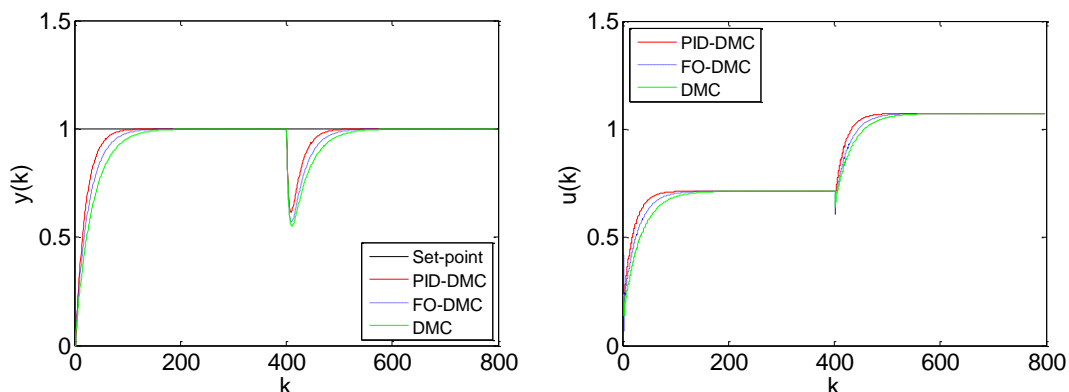


图 3.5 第一组失配情况时 PID-DMC、FO-DMC、DMC 方法控制下的系统闭环响应对比曲线

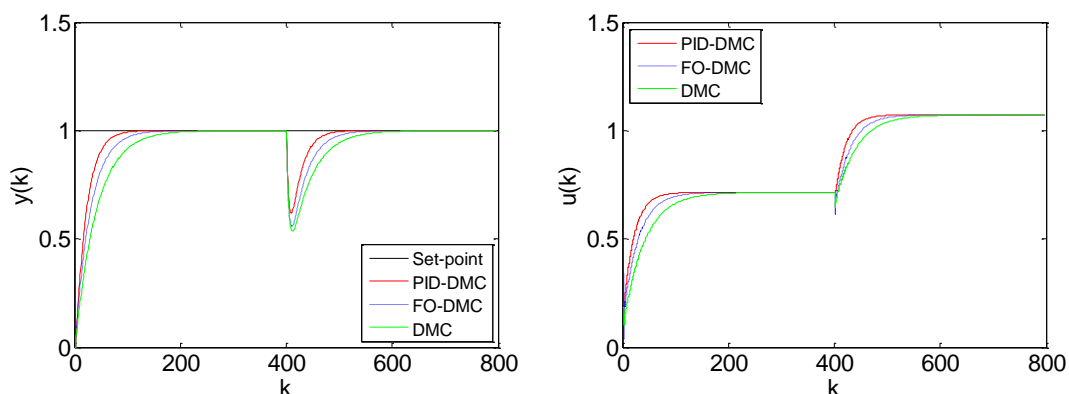


图 3.6 第二组失配情况时 PID-DMC、FO-DMC、DMC 方法控制下的系统闭环响应对比曲线

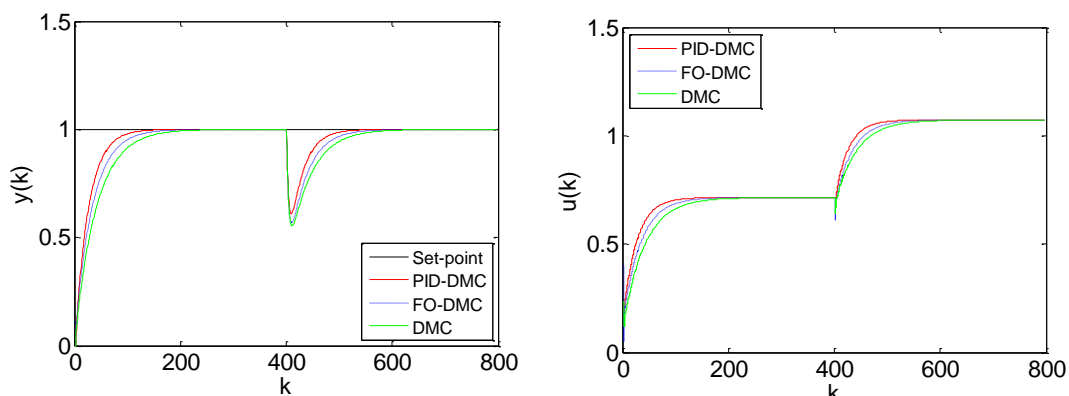


图 3.7 第三组失配情况时 PID-DMC、FO-DMC、DMC 方法控制下的系统闭环响应对比曲线

### 3.4 本章小结

本章针对传统 DMC 方法在实际工业控制中存在的不足,介绍了两种改进的 DMC 方法,进一步推广了 DMC 方法在实际过程中的应用,在保证系统良好控制性能的同时,使得系统的控制参数调节更加灵活。最后通过催化裂化过程中的稳定汽油蒸汽压力控制实例,对系统在三种方法下的控制效果进行对比分析。对比结果表明基于 PID 算子改进的 DMC 方法在实际过程中的控制性能最优。



## 第 4 章 基于分数阶系统的动态矩阵控制

### 4.1 引言

随着实际过程中控制对象的日益复杂，整数阶微分方程描述的模型往往只能体现控制系统的部分动态特性，而分数阶微积分的任意阶次数及其特有的记忆特性，使得分数阶微分方程恰能更精确的体现系统的内在特性。因而研究分数阶系统的控制问题具有极其重要的理论意义和工程价值。动态矩阵控制(DMC)方法作为预测控制算法的一种，具有对模型要求低、计算量小、处理延时的方法简单易行等特点，在实际过程中的应用极为广泛。因此本章针对一类单输入单输出分数阶系统，提出了一种分数阶系统的动态矩阵控制方法。该方法首先通过 Oustaloup 方法对分数阶系统的模型进行近似处理，然后采用基于 PID 算子改进的 DMC 方法对其实施控制，进而推进了整数阶动态矩阵控制方法在分数阶系统中的应用。最后通过高温电阻炉温度控制系统的案例研究，对依据本章方法所设计控制器的控制性能进行验证。

### 4.2 分数阶系统的 PID 型动态矩阵控制方法

本节将对分数阶系统中 PID 型动态矩阵控制(PID-DMC)算法的具体实现进行进一步描述，其所得控制器的基本特性与应用于整数阶系统中的常规 DMC 算法相同。

#### 4.2.1 分数阶模型的处理

由于绝大多数控制算法的设计都是用于整数阶系统，所以很难直接推导出适用分数阶系统的控制算法。因此本章将采用 Oustaloup 近似方法求解分数阶微分方程的数值解。

Oustaloup 近似方法表示如下：

$$s^\alpha \approx K \prod_{k=1}^N \frac{s + w_k'}{s + w_k} \quad (4.1)$$

其中， $K = w_h^\alpha$ ， $w_k' = w_b w_u^{(2k-1-\alpha)/N}$ ， $w_k = w_b w_u^{(2k-1+\alpha)/N}$ ， $w_u = \sqrt{w_h/w_b}$  且  $0 < \alpha < 1$ ， $N$  为选定的近似阶次， $w_b$ ， $w_h$  分别为拟合频率的上限和下限。

利用 Oustaloup 近似方法，将(4.1)代入(2.16)中用以描述分数阶系统的分数阶传递函数形式，可以得到近似的系统的整数阶高阶模型。在采样时间下对所得模型加零阶保持器并离散化，进一步得到过程模型的差分方程模型形式如下：

$$\begin{aligned} y(k) = & -F_1 y(k-1) - F_2 y(k-2) - \cdots - F_{L_s} y(k-L_s) \\ & + H_1 u(k-1) + H_2 u(k-2) + \cdots + H_{L_s} u(k-L_s) \end{aligned} \quad (4.2)$$

其中,  $L_s$  为离散模型的长度,  $F_j, H_j (j=1, 2, \dots, L_s)$  分别为离散近似后得到的系数。

#### 4.2.2 分数阶 PID 型动态矩阵控制器的设计

通过阶跃响应实验获取系统经过近似处理后的模型向量  $a = [a_1, a_2, \dots, a_N]^T$ , 其中  $N$  为建模时域,  $a_1, a_2, \dots, a_N$  为采样时间  $T_s$  下采样所得的阶跃响应数据。进而建立模型的动态矩阵形式如下:

$$A = \begin{bmatrix} a_1 & 0 & \cdots & 0 \\ a_2 & a_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_P & a_{P-1} & \cdots & a_{P-M+1} \end{bmatrix} \quad (4.3)$$

其中,  $A$  表示被控对象的  $P \times M$  阶动态矩阵,  $P, M$  分别表示动态矩阵控制中优化时域和控制时域的长度, 且满足  $M < P < N$ 。

在  $k-1$  时刻给过程施加控制增量  $\Delta u(k-1)$  后得到模型的输出预测值:

$$y_{N1}(k-1) = y_{N0}(k-1) + a\Delta u(k-1) \quad (4.4a)$$

其中,

$$\begin{aligned} y_{N1}(k-1) &= [y_1(k|k-1), y_1(k+1|k-1), \dots, y_1(k+N-1|k-1)]^T \\ y_{N0}(k-1) &= [y_0(k|k-1), y_0(k+1|k-1), \dots, y_0(k+N-1|k-1)]^T \end{aligned} \quad (4.4b)$$

$y_{N1}(k-1)$  表示过程对象在  $k-1$  时刻的模型预测输出向量,  $y_{N0}(k-1)$  表示  $k-1$  时刻过程对象的初始预测输出向量,  $k+i|k$  代表  $k$  时刻对  $k+i$  时刻的预测。

由于模型的不确定性使得预测输出与实际输出之间出现偏差, 因此需要对所获得的未来输出预测进行修正; 选择修正方法形式如下:

$$y_{cor}(k) = y_{N1}(k-1) + he(k) \quad (4.5a)$$

其中,

$$\begin{aligned} y_{cor}(k) &= [y_{cor}(k|k), y_{cor}(k+1|k), \dots, y_{cor}(k+N-1|k)]^T \\ e(k) &= y(k) - y_1(k|k-1) \\ h &= [1, \alpha, \dots, \alpha]^T \end{aligned} \quad (4.5b)$$

$y_{cor}(k)$  为  $k$  时刻未来输出预测的修正值,  $e(k)$  为  $k$  时刻实际过程输出和模型输出预测之间的误差值,  $y(k)$  为  $k$  时刻测得实际过程输出值,  $h$  为误差修正向量,  $\alpha$  为误差校正系数。

在  $k$  时刻需要转移  $y_{cor}(k)$  的元素形成初始的预测输出, 转移过程描述如下:

$$y_{N0}(k) = S y_{cor}(k) \quad (4.6a)$$

其中,  $S$  为  $N \times N$  阶的状态转移矩阵,

$$S = \begin{bmatrix} 0 & 1 & 0 & \cdots & \cdots \\ 0 & 0 & 1 & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots & \cdots \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix} \quad (4.6b)$$

由于存在模型截断， $y_{N0}(k)$ 的最后一个元素 $y_0(k+N|k)$ 将被 $y_{cor}(k+N-1|k)$ 替代。

在获得 $k$ 时刻的初始输出预测之后，加入 $M$ 个连续的控制增量 $\Delta u_M(k) = [\Delta u(k), \Delta u(k+1), \dots, \Delta u(k+M-1)]^T$ 得到未来模型的输出预测形式如下：

$$y_{PM}(k) = y_{P0}(k) + A\Delta u_M(k) \quad (4.7a)$$

其中，

$$\begin{aligned} y_{PM}(k) &= [y_M(k+1|k), y_M(k+2|k), \dots, y_M(k+M|k)]^T \\ y_{P0}(k) &= [y_0(k+1|k), y_0(k+2|k), \dots, y_0(k+M|k)]^T \end{aligned} \quad (4.7b)$$

$y_{PM}(k)$ 为 $k$ 时刻在 $\Delta u_M(k)$ 作用下的未来输出预测， $y_{P0}(k)$ 为 $k$ 时刻的初始输出预测。

选取分数阶动态矩阵控制的参考轨迹及目标函数<sup>[79]</sup>

$$J = E_0(k)^T K_i E_0(k) + \Delta E_0(k)^T K_p \Delta E_0(k) + \Delta^2 E_0(k)^T K_d \Delta^2 E_0(k) + \Delta u_M(k)^T R \Delta u_M(k) \quad (4.8a)$$

其中，

$$\begin{aligned} E_0(k) &= Y_r(k) - y_{PM}(k) = [e_0(k+1), e_0(k+2), \dots, e_0(k+P)]^T \\ \Delta E_0(k) &= \Delta Y_r(k) - \Delta y_{PM}(k) = [\Delta e_0(k+1), \Delta e_0(k+2), \dots, \Delta e_0(k+P)]^T \end{aligned} \quad (4.8b)$$

$$\begin{aligned} \Delta^2 E_0(k) &= \Delta^2 Y_r(k) - \Delta^2 y_{PM}(k) = [\Delta^2 e_0(k+1), \Delta^2 e_0(k+2), \dots, \Delta^2 e_0(k+P)]^T \\ Y_r(k) &= [y_r(k+1), y_r(k+2), \dots, y_r(k+P)]^T \\ y_r(k+i) &= \lambda^i y(k) + (1-\lambda^i)c(k) \end{aligned} \quad (4.8c)$$

$E_0(k)$ 表示过程对象的输出误差， $\Delta$ 表示差分算子， $K_p = \text{diag}(k_p, \dots, k_p)$ ， $K_i = \text{diag}(k_i, \dots, k_i)$ ， $K_d = \text{diag}(k_d, \dots, k_d)$ 分别为比例系数矩阵、积分系数矩阵、微分系数矩阵； $R = \text{diag}(r_1, r_2, \dots, r_p)$ 为控制加权系数矩阵。 $y_r(k+i)$ 为给定期望输出的参考轨迹， $y(k)$ 为 $k$ 时刻过程实测输出， $c(k)$ 为 $k$ 时刻的期望输出， $\lambda$ 为参考轨迹柔化因子。

由式(4.8b)有：

$$\begin{aligned} \Delta e_0(k+i) &= \Delta y_r(k+i) - \Delta y_{N0}(k+i|k) \\ &= y_r(k+i) - y_{N0}(k+i|k) - (y_r(k+i-1) - y_{N0}(k+i-1|k)) \\ &= e_0(k+i) - e_0(k+i-1) \end{aligned} \quad (4.9a)$$

同理可得

$$\Delta^2 e_0(k+i) = \Delta e_0(k+i) - \Delta e_0(k+i-1) \quad (4.9b)$$

为方便起见，特别定义  $e_0(k) = \Delta e_0(k) = 0$

并引入一个矩阵

$$S_1 = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ -1 & 1 & 0 & \cdots & 0 \\ 0 & -1 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & -1 & 1 \end{bmatrix}_{P \times P} \quad (4.10)$$

进而有

$$\begin{cases} \Delta E_0(k) = S_1 E_0(k) \\ \Delta^2 E_0(k) = S_1 \Delta E_0(k) = S_1^2 E_0(k) \end{cases} \quad (4.11)$$

将(4.11)代入(4.8a)，将目标函数进一步转化为

$$\begin{aligned} J &= E_0(k)^T K_i E_0(k) + E_0(k)^T S_1^T K_p S_1 E_0(k) \\ &\quad + E_0(k)^T (S_1^2)^T K_d (S_1^2) E_0(k) + \Delta u_M(k)^T R \Delta u_M(k) \\ &= E_0(k)^T Q E_0(k) + \Delta u_M(k)^T R \Delta u_M(k) \end{aligned} \quad (4.12)$$

其中， $Q = K_i + K_p S_1^T S_1 + K_d (S_1^2)^T (S_1^2)$

以  $\Delta u_M(k)$  为控制变量，通过求解目标函数(4.12)的极值即令  $\frac{\partial J}{\partial \Delta u_M(k)} = 0$ ，

得到如下形式的最优控制律：

$$\Delta u_M(k) = (R + A^T Q A)^{-1} A^T Q (Y_r(k) - y_{p0}(k)) \quad (4.13)$$

令  $D^T = [1, 0, \dots, 0]_{1 \times P}$ ，即可得到被控对象所需的实际控制增量  $\Delta u(k)$ ：

$$\Delta u(k) = D^T (R + A^T Q A)^{-1} A^T Q (Y_r(k) - y_{p0}(k)) \quad (4.14)$$

### 4.3 案例研究：加热炉温度控制系统

本节将 FO-PIDDMC 方法运用于一个实际的加热炉温度控制系统，并通过 MATLAB 仿真验证了其性能。其中图 4.1 中给出了加热炉的具体工艺流程，图 4.2 通过加热炉对象的实验数据对比了实际系统与模型的阶跃响应曲线，图 4.3-图 4.6 分别给出了模型匹配及失配情况下 FO-PIDDMC 与传统整数阶控制方法的对比效果，并对本章所提控制方法在各方面性能上的优势做出了说明。下面将通过图 4.1 所示的工艺流程图，对加热炉的内部构造及实现方式进行说明。

#### 4.3.1 加热炉工艺流程介绍

图 4.1 所显示的是基于 SXF-4-10 高温电阻炉的温度控制系统的工艺流程，其中高温炉的温度控制手段是采用控制固态继电器的通断来实现的，炉温的期望值可在计算机中预先设定。

首先通过 K 分度热电偶来检测 SXF-4-10 高温炉的实际炉温，再将其转化为相应的电压信号。由于 PCI-1802LU CR 信号采集卡无法识别过小的电压信号，

因此要将其先通过 SG-3011 CR 信号放大器放大, 再由 DN-37CR 端子板通过 DN-3710 CR 传送至工控机卡槽里的 PCI-1802LU CR 信号采集卡, 进而通过计算机将该信号采集卡采集所得模拟电压信号转换为相应的温度值。当计算机输入数字信号 0x01 至 PIO-D24U 信号输出卡时, 会使其输出一个 +5V 电压, 再通过 DN-3710 CR 电缆线和 DN-37CR 端子板给 SSR-380D40 固态继电器输入该电压, 此时的固态继电器左侧将导通, 高温炉呈现为加热状态。当计算机输入数字信号 0x00 至 PIO-D24U 信号输出卡时, 会使该信号输出卡输出一个 0V 电压, 再通过 DN-3710 CR 电缆线和 DN-37CR 端子板给固态继电器输入该电压, 此时的固态继电器左侧会断开, 进而使高温炉的加热回路断开, 呈现为散热状态。

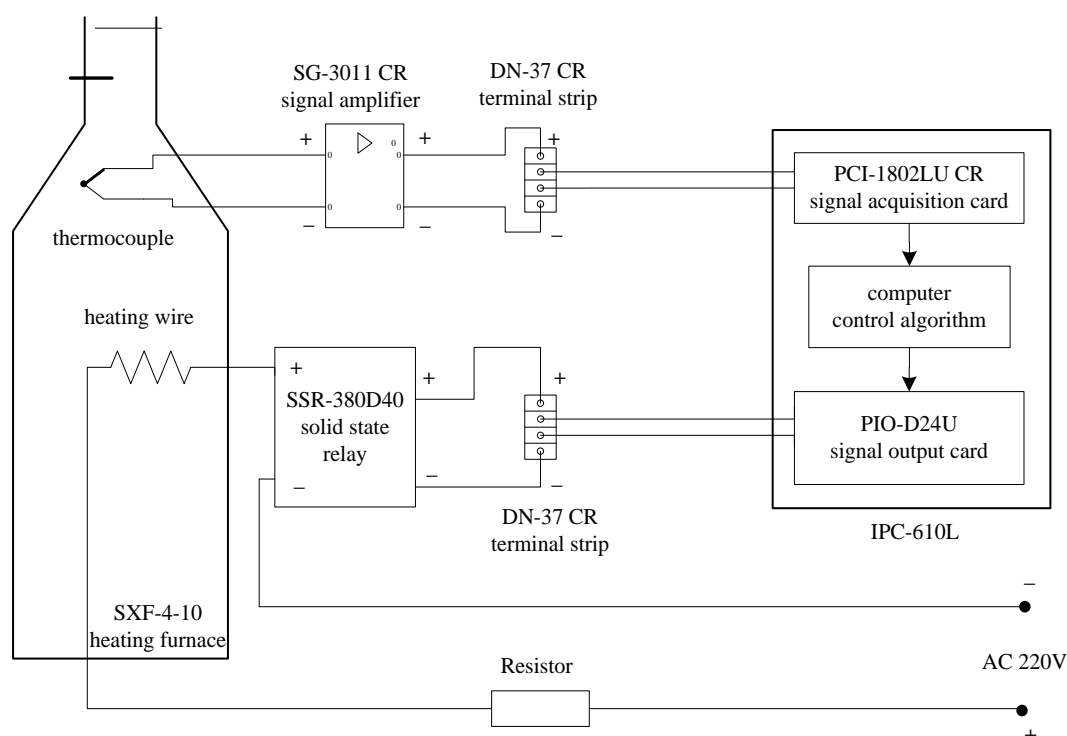


图4.1 SXF-4-10 高温电阻炉温度控制系统

### 4.3.2 模型对比与分析

将本章所提的分数阶 PID 型动态矩阵控制 (FO-PIDDMC) 方法运用到加热炉温度控制的实例中, 通过 MATLAB 仿真对所设计控制器的性能进行验证。加热炉的温度控制的调节手段, 是通过控制占空比进而调节单个控制周期内的加热时间来实现的。对加热炉输入一个阶跃信号, 通过阶跃响应实验记录其温度输出值, 可以建立加热炉温度对象的分数阶模型, 同时使用两点法可建立其一阶加纯滞后模型。

由采集到的加热炉实时温度数据, 得到加热炉温度对象的分数阶模型和一阶加纯滞后模型分别为:

$$G(s) = \frac{0.585}{300s^{0.92} + 1} e^{-240s} \quad (4.15a)$$

$$G(s) = \frac{0.57}{500s+1} e^{-240s} \quad (4.15b)$$

通过 Oustaloup 方法将式(4.15a)中的分数阶模型近似为如下形式的高阶模型:

$$G(s) = \frac{0.585s^4 + 478.2s^3 + 2186s^2 + 56.18s + 0.008075}{1.726 \times 10^5 s^4 + 1.202 \times 10^6 s^3 + 5.047 \times 10^4 s^2 + 106.3s + 0.01382} e^{-240s} \quad (4.16)$$

其中, 选取微分算子的近似阶次为  $N=4$ , Oustaloup 近似的上下限拟合频率分别为  $w_h = 10^3$  和  $w_b = 10^{-6}$ 。

对实际采集的过程对象温度数据同所得三种模型的输出值进行对比, 求取其标准差如下表 4.1 所示:

表4.1 三种模型输出值的方差对比结果

模型 $G(s)$	标准差 SE
分数阶模型	1.4856
一阶加纯滞后模型	1.7592
Oustaloup 近似模型	1.5000

将阶跃响应实验采集所得加热炉温度数据与模型的阶跃响应输出在 MATLAB 中进行对比, 对比曲线如下图所示:

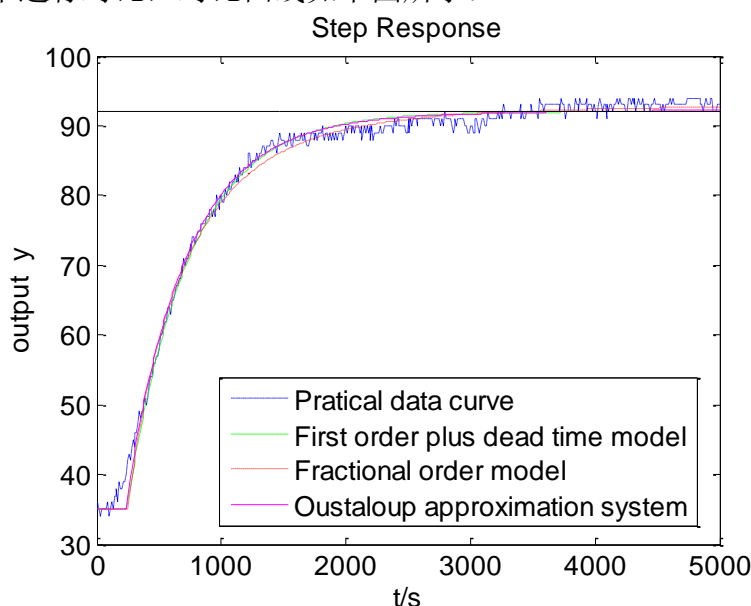


图4.2 实际系统与模型的阶跃响应曲线

从图 4.2 中可以发现, 输出响应最接近实际过程的是分数阶模型, 其次是 Oustaloup 近似模型, 而运用两点法所得一阶加纯滞后模型的相对误差较大。

本文选取的采样时间为  $T_s = 40$ , 对(4.15b)中模型加零阶保持器并离散化, 得到如下形式的离散模型:

$$G(z) = \frac{0.04382}{z^7 - 0.9231z^6} \quad (4.17)$$

同时对(4.16)中模型加零阶保持器并离散化, 得到其离散模型形式如下:

$$G(z) = \frac{3.389 \times 10^{-6} z^4 + 0.05727 z^3 - 0.07843 z^2 + 0.02139 z - 2.48 \times 10^{-5}}{z^4 - 2.117 z^3 + 1.302 z^2 - 0.1847 z} z^{(-6)} \quad (4.18)$$

### 4.3.3 仿真结果及分析

为验证本文所提方法的有效性, 将所设计的分数阶 PID 动态矩阵控制方法同常规整数阶动态矩阵方法通过 MATLAB 仿真进行对比分析。其中分数阶 PID-DMC 是以式(4.18)中所示的 Oustaloup 近似模型作为其标称模型来设计控制器的, 而整数阶 DMC 则以式(4.17)中一阶加纯滞后模型为其标称模型设计控制器, 最后将二者所得最优控制律均作用于 Oustaloup 近似模型所描述的分数阶系统。在仿真过程中设定值选取  $c(k) = 100$ , 并在  $k = 400$  时刻持续加入幅值为  $-10$  的输出干扰, 分别考虑模型匹配及失配情况下的控制效果。表 4.2 中列出了两种控制器的其他参数详情。

表 4.2 FO-PIDDMC 及 DMC 方法的参数表

Parameters	FO-PIDDMC	DMC
$P$	8	
$M$	3	
$N$	100	
$T_s$	40	
$\alpha$	1	
$\lambda$	0.95	
$q_i$	\	1
$r_i$	0.8	
$k_p$	20	\
$k_i$	0.5	\
$k_d$	5.6	\

#### (1) 模型匹配

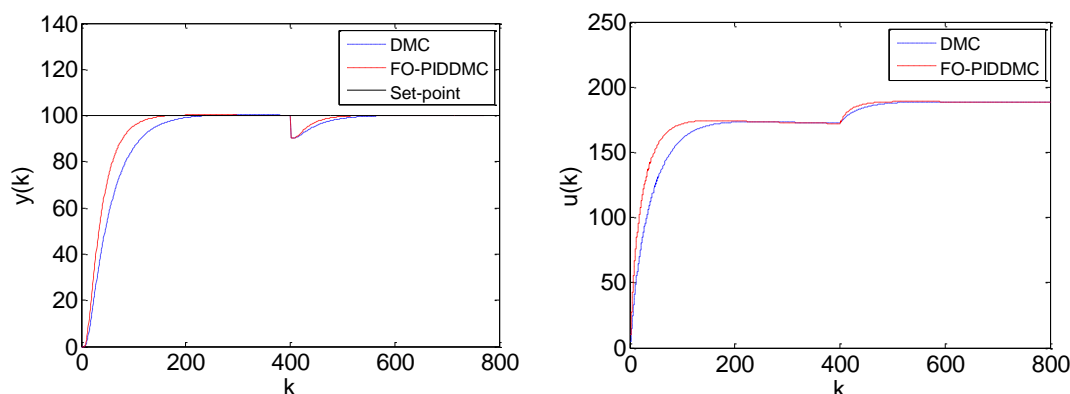


图4.3 模型匹配时FO-PIDDMC、DMC方法控制下的系统闭环响应对比曲线

图 4.3 显示了模型匹配情况下被控系统在分数阶控制器和整数阶控制器控

制下输出响应对比曲线。从图中可以发现，分数阶 PID-DMC 和整数阶 DMC 所控制的分数阶系统均能获得稳定控制。相比之下，系统在分数阶 PID-DMC 控制下的响应曲线不仅没有振荡和超调，并且其响应曲线更加平滑，拥有良好的跟踪性能、较快的上升速度及较强的抗干扰能力。

## (2) 模型失配

在实际过程中，往往会存在一些不确定因素或不可测的干扰，使得建立的模型存在一定误差，因而导致了模型可能会出现失配的情况。接下来使用蒙特卡洛方法对式(4.15a)中的分数阶模型参数随机产生三组失配 30% 的参数，得到如下形式的三组失配模型，并将其用来替换实际的过程对象。

$$\text{第一组: } G(s) = \frac{0.7253}{299.3794s^{1.0971}+1}e^{-2.60.8s} \quad (4.19a)$$

$$\text{第二组: } G(s) = \frac{0.5564}{293.8790s^{1.1141}+1}e^{-2.43.6s} \quad (4.19b)$$

$$\text{第三组: } G(s) = \frac{0.6412}{357.2354s^{0.8371}+1}e^{-2.09.7s} \quad (4.19c)$$

同样在相同条件下对比整数阶控制器和分数阶控制器控制下的系统响应曲线，可以发现二者在模型失配情况下获得的控制性能仍较为稳定。图 4.4-图 4.6 分别给出了控制系统在三组模型失配情况下的闭环响应对比曲线。通过三组模型失配情况下的对比曲线，可以发现分数阶 PID-DMC 控制器获得的控制效果均比整数阶 DMC 控制器好。其中，图 4.4 和图 4.5 中的仿真效果显示分数阶 PID-DMC 控制器拥有更快的上升速度及更强的抗干扰能力，由图 4.6 可以看出整数阶控制器的响应速度较慢。总之，相比之下，本文所提分数阶控制器拥有更强的抗干扰能力及跟踪性能，因而获得的控制效果更好。

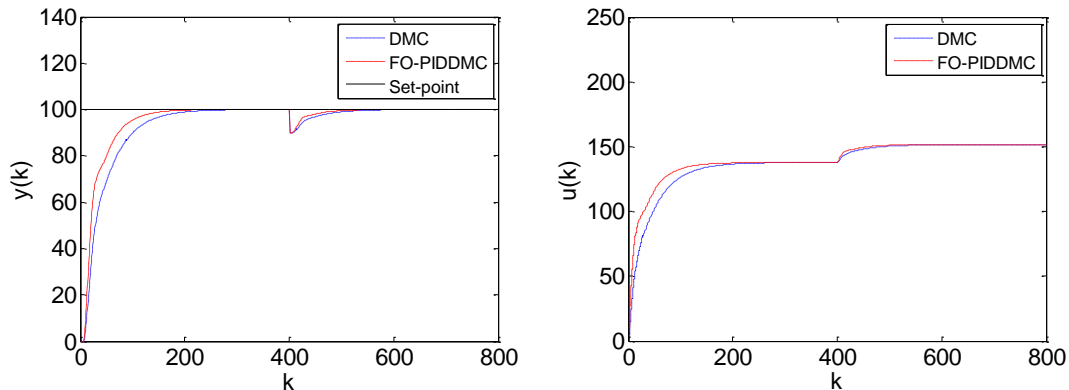


图4.4 第一组失配情况下FO-PIDDMC、DMC方法控制下系统的闭环响应对比曲线



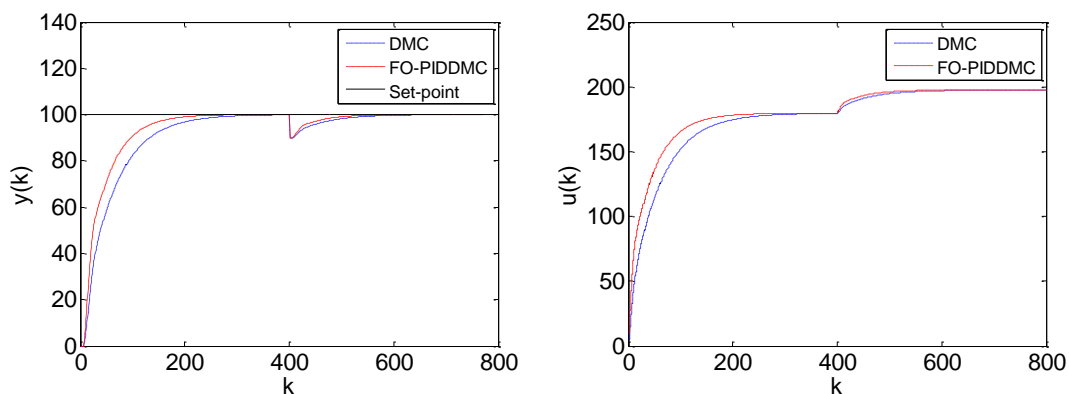


图4.5 第二组失配情况下FO-PIDDMC、DMC方法控制下系统的闭环响应对比曲线

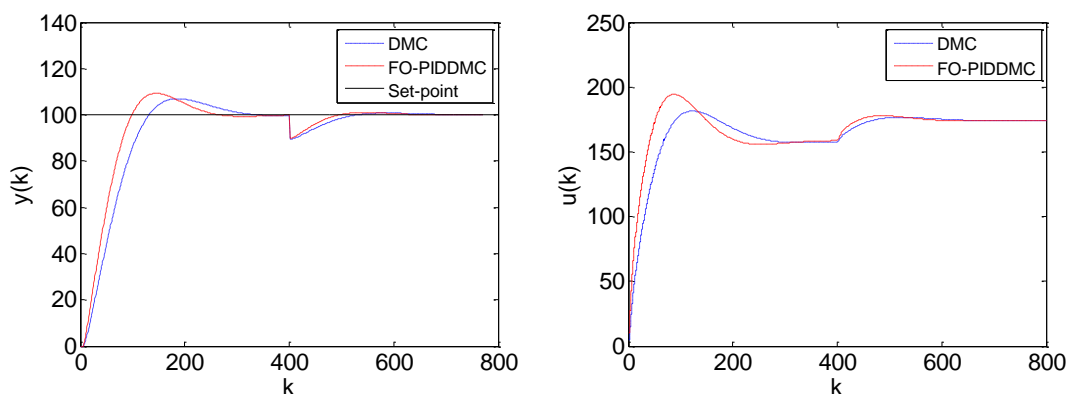


图4.6 第三组失配情况下FO-PIDDMC、DMC方法控制下系统的闭环响应对比曲线

#### 4.4 本章小结

本章提出了一种基于分数阶系统的 PID 型动态矩阵控制器的设计方法，该方法通过将基于 PID 算子改进的 DMC 方法引入到分数阶线性系统的控制中，进一步推广了动态矩阵控制方法在分数阶系统中的应用，并通过一个加热炉温度控制系统的仿真实例验证了本章所提方法的有效性和可行性。

## 第5章 基于分数阶系统的分布式动态矩阵控制

### 5.1 引言

实际过程中往往存在着大量复杂高维的大规模系统，其变量、约束的数目通常有几十个或上百个之多，因而针对这样一类大规模预测控制的在线实施问题便显得尤为重要。为更精确、全面的描述此类系统的内在特性，本章将考虑采用一类多入多出分数阶模型来描述该系统。随着计算机网络技术的迅速发展，控制作为信息加工的一种操作方式，已逐步由集中式控制结构转变为分布式控制结构。分布式动态矩阵控制作为预测控制在分布式控制结构中的典型应用，综合利用网络通信技术与控制理论，把一个复杂高维系统的在线优化问题分散到各个低维子系统中分布求解，有效降低了问题的规模与复杂性，能很好的控制存在多变量、强耦合的被控对象。因此本章通过在一类多变量分数阶系统中应用一种基于 PID 算子改进的分布式动态矩阵控制，进而得到基于分数阶系统的分布式动态矩阵控制方法。基于该方法设计的控制器不仅继承了分布式动态矩阵控制的优良性能，同时还拥有 PID 控制结构简单且易于调整的优势。

### 5.2 分数阶系统的分布式动态矩阵控制方法

#### 5.2.1 分布式动态矩阵控制中预测模型的计算

将工业过程中的多变量  $N$  输入  $N$  输出大规模系统看成由  $N$  个一阶惯性加纯滞后(FOPDT)模型的分数阶子系统组成，可以得到如下形式的  $N$  输入  $N$  输出多变量分数阶系统：

$$G(s) = \begin{bmatrix} \frac{K_{11}e^{-\tau_{11}s}}{T_{11}s^{\alpha_{11}} + 1} & \frac{K_{12}e^{-\tau_{12}s}}{T_{12}s^{\alpha_{12}} + 1} & \dots & \frac{K_{1n}e^{-\tau_{1n}s}}{T_{1n}s^{\alpha_{1n}} + 1} \\ \frac{K_{21}e^{-\tau_{21}s}}{T_{21}s^{\alpha_{21}} + 1} & \frac{K_{22}e^{-\tau_{22}s}}{T_{22}s^{\alpha_{22}} + 1} & \dots & \frac{K_{2n}e^{-\tau_{2n}s}}{T_{2n}s^{\alpha_{2n}} + 1} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{K_{n1}e^{-\tau_{n1}s}}{T_{n1}s^{\alpha_{n1}} + 1} & \frac{K_{n2}e^{-\tau_{n2}s}}{T_{n2}s^{\alpha_{n2}} + 1} & \dots & \frac{K_{nn}e^{-\tau_{nn}s}}{T_{nn}s^{\alpha_{nn}} + 1} \end{bmatrix} \quad (5.1)$$

其中， $K_{ij}$  为多变量过程对象第  $j(1 \leq j \leq n)$  个输入对第  $i(1 \leq i \leq n)$  个输出的稳态增益， $T_{ij}$  为多变量过程对象第  $j$  个输入对第  $i$  个输出的时间常数， $\tau_{ij}$  为多变量过程对象第  $j$  个输入对第  $i$  个输出的滞后时间， $\alpha_{ij}$  表示多变量过程对象第  $j$  个输入对第  $i$  个输出的分数阶阶次。

根据分布式预测控制思想，将一个  $N$  输入  $N$  输出大规模分数阶系统分散为

$N$  个智能体子系统；

利用式(4.1)中描述的 Oustaloup 近似方法对式 (5.1) 中的多变量分数阶系统进行近似处理, 然后给经过处理得到的 Oustaloup 近似模型输入一个阶跃信号, 并记录该近似模型的阶跃响应曲线。

对所得阶跃响应曲线做滤波处理, 然后拟合成一条光滑曲线, 记录光滑曲线上的每个采样时刻对应的阶跃响应数据, 第一个采样时刻为  $T_s$ , 采样时刻顺序为  $T_s, 2T_s, 3T_s, \dots$ ; 近似模型的阶跃响应将在某一个时刻  $t_L = L_j T_s$  后趋于平稳, 当  $a_{ij}(t)(t > L_j)$  与  $a_{ij}(L_j)$  的误差和测量误差有相同的数量级时, 即可认为  $a_{ij}(L_j)$  近似等于阶跃响应的稳态值。建立第  $j$  个输入对第  $i$  个输出之间的阶跃响应模型向量  $\mathbf{a}_{ij}$  :

$$\mathbf{a}_{ij} = [a_{ij}(1), a_{ij}(2), \dots, a_{ij}(L_j)]^T \quad (5.2)$$

其中,  $a_{ij}(k)$  为  $t = kT_s$  的阶跃响应采样值,  $L_j$  为第  $j$  个输入对第  $i$  个输出的建模时域,  $T$  为矩阵的转置符号。

利用获取的模型向量  $\mathbf{a}_{ij}$  建立多变量过程对象的动态矩阵, 其形式如下:

$$A_{ij} = \begin{bmatrix} a_{ij}(1) & 0 & \cdots & 0 \\ a_{ij}(2) & a_{ij}(1) & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ a_{ij}(M) & a_{ij}(M-1) & \cdots & a_{ij}(1) \\ \vdots & \vdots & & \vdots \\ a_{ij}(P) & a_{ij}(P-1) & \cdots & a_{ij}(P+M-1) \end{bmatrix}, A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1N} \\ A_{21} & A_{22} & \cdots & A_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ A_{N1} & A_{N2} & \cdots & A_{NN} \end{bmatrix} \quad (5.3)$$

其中,  $A_{ij}$  为第  $j$  个智能体输入对第  $i$  个智能体输出的  $P \times M$  阶动态矩阵,  $a_{ij}(k)$  为第  $j$  个输入对第  $i$  个输出阶跃响应的数据,  $P$ 、 $M$  分别为分布式动态矩阵控制中优化时域和控制时域的长度, 令  $L_j = L$ , 且满足  $M < P < L$ 。

在  $k-1$  时刻加入各智能体的控制增量  $\Delta u_1(k-1), \Delta u_2(k-1), \dots, \Delta u_n(k-1)$ , 得到第  $i$  个智能体的模型预测值  $y_{i,P}(k-1)$  :

$$y_{i,P}(k-1) = y_{i,0}(k-1) + A_{ii,0}\Delta u_i(k-1) + \sum_{j=1, j \neq i}^n A_{ij,0}\Delta u_j(k-1) \quad (5.4)$$

其中,

$$y_{i,P}(k-1) = [y_{i,1}(k|k-1), y_{i,1}(k+1|k-1), \dots, y_{i,1}(k+L-1|k-1)]^T$$

$$y_{i,0}(k-1) = [y_{i,0}(k|k-1), y_{i,0}(k+1|k-1), \dots, y_{i,0}(k+L-1|k-1)]^T$$

$$A_{ii,0} = [a_{ii}(1), a_{ii}(2), \dots, a_{ii}(L)]^T, A_{ij,0} = [a_{ij}(1), a_{ij}(2), \dots, a_{ij}(L)]^T$$

$y_{i,1}(k|k-1), y_{i,1}(k+1|k-1), \dots, y_{i,1}(k+L-1|k-1)$  分别表示第  $i$  个智能体在  $k-1$  时刻对  $k, +k \cdots 1, +k$  时刻的模型预测值,  $y_{i,0}(k|k-1), y_{i,0}(k+1|k-1), \dots, y_{i,0}(k+L-1|k-1)$  表示第  $i$  个智能体在  $k-1$  时刻对

$k, k+1, \dots, k+L-1$ 时刻的初始预测值,  $A_{i,0}, A_{j,0}$  分别为第  $i$  个智能体和第  $j$  个智能体对第  $i$  个智能体的阶跃响应数据建立的矩阵,  $\Delta u_1(k-1), \Delta u_2(k-1), \dots, \Delta u_n(k-1)$  分别表示  $k-1$  时刻各智能体的输入控制增量;

在通过检测得到第  $i$  个智能体在  $k$  时刻的实际输出值  $y_i(k)$  后, 为补偿实际过程中由模型失配、环境扰动等未知因素引起的预测输出与实际输出的偏差, 这里采取加权方式利用输出误差信息对第  $i$  个智能体未来的输出预测值进行修正, 具体修正方法如下;

$$y_{i,cor}(k) = y_{i,0}(k-1) + h^* e_i(k) \quad (5.5)$$

其中,

$$\begin{aligned} y_{i,cor}(k) &= [y_{i,cor}(k|k), y_{i,cor}(k+1|k), \dots, y_{i,cor}(k+L-1|k)]^T \\ e_i(k) &= y_i(k) - y_{i,1}(k|k-1) \\ h &= [1, \alpha, \dots, \alpha]^T \end{aligned}$$

$e_i(k)$  表示  $k$  时刻第  $i$  个智能体的模型预测误差值,  $y_i(k)$  表示  $k$  时刻测得的第  $i$  个智能体的实际输出值,  $y_{i,cor}(k)$  表示  $k$  时刻修正后的模型输出值,  $y_{i,cor}(k|k), y_{i,cor}(k+1|k), \dots, y_{i,cor}(k+L-1|k)$  分别表示第  $i$  个智能体在  $k$  时刻模型的修正值,  $h$  为误差补偿的权矩阵,  $\alpha$  为误差校正系数;

由于时间基点在  $k$  时刻会产生变化, 即预测的未来时间点将会移动至  $k+1, k+2, \dots, k+N$ , 因而需要对第  $i$  个智能体修正后的模型输出值进行移位操作才能得到  $k$  时刻第  $i$  个智能体模型预测的初始响应值  $y_{i,0}(k)$ :

$$y_{i,0}(k) = S y_{i,cor}(k) \quad (5.6)$$

其中,

$$S = \begin{bmatrix} 0 & 1 & 0 & \dots & \dots \\ 0 & 0 & 1 & 0 & \dots \\ \vdots & \vdots & \vdots & \ddots & \dots \\ 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & \dots & 0 & 1 \end{bmatrix}$$

$S$  为  $L \times L$  阶的状态转移矩阵, 当系统满足渐进稳定时, 有  $y_{i,cor}(k+L|k)$  等于  $y_{i,cor}(k+L-1|k)$  成立。

第  $i$  个智能体在  $M$  个连续的输入控制增量  $\Delta u_i(k), \Delta u_i(k+1), \dots, \Delta u_i(k+M-1)$  下的预测输出值  $y_{i,PM}$  为:

$$y_{i,PM}(k) = y_{i,P0}(k) + A_{ii} \Delta u_{i,M}(k) + \sum_{j=1, j \neq i}^n A_{ij} \Delta u_{j,M}(k) \quad (5.7)$$

其中,

$$y_{i,PM}(k) = [y_{i,M}(k+1|k), y_{i,M}(k+2|k), \dots, y_{i,M}(k+P|k)]^T$$

$$y_{i,P0}(k) = [y_{i,0}(k+1|k), y_{i,0}(k+2|k), \dots, y_{i,0}(k+P|k)]^T$$

$$\Delta u_{i,M}(k) = [\Delta u_i(k), \Delta u_i(k+1), \dots, \Delta u_i(k+M-1)]^T$$

$$\Delta u_{j,M}(k) = [\Delta u_j(k), \Delta u_j(k+1), \dots, \Delta u_j(k+M-1)]^T$$

$y_{i,M}(k+1|k), y_{i,M}(k+2|k), \dots, y_{i,M}(k+P|k)$  分别表示第  $i$  个智能体在  $k$  时刻对  $k+1, k+2, \dots, k+P$  时刻的模型预测输出值,  $y_{i,P0}(k)$  为  $y_{i,0}(k)$  的前  $P$  项,  $y_{i,0}(k+1|k), y_{i,0}(k+2|k), \dots, y_{i,0}(k+P|k)$  分别表示第  $i$  个智能体  $k$  时刻对  $k+1, k+2, \dots, k+P$  时刻的模型预测初始值;

### 5.2.2 分布式动态矩阵控制性能指标的选取

在分布式动态矩阵控制的性能指标中引入 PID 算子, 选取第  $i$  个智能子系统的性能指标, 其形式如下:

$$\begin{aligned} \min \tilde{J}_i(k) = & E_0^i(k)^T K_I^i E_0^i(k) + \Delta E_0^i(k)^T K_p^i \Delta E_0^i(k) + \Delta^2 E_0^i(k)^T K_d^i \Delta^2 E_0^i(k) \\ & + \Delta u_{i,M}(k)^T R_i \Delta u_{i,M}(k) \end{aligned} \quad (5.8)$$

其中,

$$E_0^i(k) = w_i(k) - y_{i,PM}(k) = [e_0^i(k+1), e_0^i(k+2), \dots, e_0^i(k+P)]^T$$

$$\Delta E_0^i(k) = \Delta w_i(k) - \Delta y_{i,PM}(k) = [\Delta e_0^i(k+1), \Delta e_0^i(k+2), \dots, \Delta e_0^i(k+P)]^T$$

$$\Delta^2 E_0^i(k) = \Delta^2 w_i(k) - \Delta^2 y_{i,PM}(k) = [\Delta^2 e_0^i(k+1), \Delta^2 e_0^i(k+2), \dots, \Delta^2 e_0^i(k+P)]^T$$

$$w_i(k) = [\omega_i(k+1), \omega_i(k+2), \dots, \omega_i(k+P)]^T$$

$$\omega_i(k+\varepsilon) = \lambda^\varepsilon y_i(k) + (1-\lambda^\varepsilon)c(k) (\varepsilon=1, 2, \dots, P)$$

$E_0^i(k)$  表示第  $i$  个智能体的输出误差,  $\Delta$  表示差分算子,  $K_p^i = \text{diag}(k_p^i, \dots, k_p^i)$ ,  $K_I^i = \text{diag}(k_I^i, \dots, k_I^i)$ ,  $K_d^i = \text{diag}(k_d^i, \dots, k_d^i)$  分别表示第  $i$  个智能体的比例系数矩阵、积分系数矩阵、微分系数矩阵;  $R_i = \text{diag}(r_1^i, r_2^i, \dots, r_p^i)$  为第  $i$  个智能体的控制加权系数矩阵,  $\omega_i(k+\varepsilon)$  为第  $i$  个智能体给定期望输出的参考轨迹,  $y_i(k)$  为  $k$  时刻第  $i$  个智能体的过程实际输出,  $c(k)$  为  $k$  时刻第  $i$  个智能体的期望输出,  $\lambda$  为参考轨迹柔化因子。

在上式中有

$$\begin{aligned} \Delta e_0^i(k+\varepsilon) &= \Delta \omega_i(k+\varepsilon) - \Delta y_{i,M}(k+\varepsilon|k) \\ &= \omega_i(k+\varepsilon) - y_{i,M}(k+\varepsilon|k) - (\omega_i(k+\varepsilon-1) - y_{i,M}(k+\varepsilon-1|k)) \end{aligned} \quad (5.9a)$$

$$\begin{aligned} &= e_0^i(k+\varepsilon) - e_0^i(k+\varepsilon-1) \\ \Delta^2 e_0^i(k+\varepsilon) &= \Delta e_0^i(k+\varepsilon) - \Delta e_0^i(k+\varepsilon-1) \end{aligned} \quad (5.9b)$$

其中,  $e_0^i(k) = \Delta e_0^i(k) = 0$

引入矩阵

$$S_1 = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ -1 & 1 & 0 & \cdots & 0 \\ 0 & -1 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & -1 & 1 \end{bmatrix}_{P \times P} \quad (5.10)$$

进而得到

$$\begin{cases} \Delta E_0^i(k) = S_1 E_0^i(k) \\ \Delta^2 E_0^i(k) = S_1 \Delta E_0^i(k) = S_1^2 E_0^i(k) \end{cases} \quad (5.11)$$

综合上述步骤可以得到第  $i$  个智能体的性能指标:

$$\begin{aligned} \min J_i(k) &= E_0^i(k)^T K_l^i E_0^i(k) + E_0^i(k)^T S_1^T K_p^i S_1 E_0^i(k) \\ &\quad + E_0^i(k)^T (S_1^2)^T K_d^i (S_1^2) E_0^i(k) + \Delta u_{i,M}(k)^T R_i \Delta u_{i,M}(k) \\ &= E_0^i(k)^T Q_i E_0^i(k) + \Delta u_{i,M}(k)^T R_i \Delta u_{i,M}(k) \end{aligned} \quad (5.12)$$

其中,  $Q_i = K_l^i + K_p^i S_1^T S_1 + K_d^i (S_1^2)^T (S_1^2)$

### 5.2.3 分数阶分布式动态矩阵控制器的设计

依据纳什最优的思想, 以  $\Delta u_{i,M}(k)$  为控制变量, 最小化式(5.12)中目标函数

即求解  $\frac{\partial J_i(k)}{\partial \Delta u_{i,M}(k)} = 0$ , 得到如下形式的最优控制律:

$$\Delta u_{i,M}^*(k) = D_{ii}(w_i(k) - y_{i,P0}(k) - \sum_{j=1, j \neq i}^n A_{ij} u_{j,M}^*(k)) \quad (5.13a)$$

$$\Delta u_i^*(k) = [1, 0, \dots, 0] \Delta u_{i,M}^*(k) \quad (5.13b)$$

其中,  $D_{ii} = (R_i + A_{ii}^T Q_i A_{ii})^{-1} A_{ii}^T Q_i$

进一步可以得到  $k$  时刻第  $i$  个智能体的新一轮迭代最优解为:

$$\Delta u_{i,M}^{l+1}(k) = D_{ii}(w_i(k) - y_{i,P0}(k) - \sum_{j=1, j \neq i}^n A_{ij} \Delta u_{j,M}^l(k)) \quad (5.14)$$

整个系统在  $k$  时刻的最优控制律:

$$\Delta u_M^{l+1}(k) = D_1(w(k) - y_{P0}(k)) - D_0 \Delta u_M^l(k) \quad (5.15)$$

其中,

$$\begin{aligned} \Delta u_M^{l+1}(k) &= [\Delta u_{1,M}^{l+1}(k), \Delta u_{2,M}^{l+1}(k), \dots, \Delta u_{n,M}^{l+1}(k)]^T \\ \Delta u_M^l(k) &= [\Delta u_{1,M}^l(k), \Delta u_{2,M}^l(k), \dots, \Delta u_{n,M}^l(k)]^T \\ w(k) &= [w_1(k), w_2(k), \dots, w_n(k)]^T, \quad y_{P0}(k) = [y_{1,P0}(k), y_{2,P0}(k), \dots, y_{n,P0}(k)]^T \\ D_1 &= \begin{bmatrix} D_{11} & 0 & \cdots & 0 \\ 0 & D_{22} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & D_{NN} \end{bmatrix}, \quad D_0 = \begin{bmatrix} 0 & -D_{11}A_{12} & \cdots & -D_{11}A_{1N} \\ -D_{22}A_{21} & 0 & \cdots & -D_{22}A_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ -D_{NN}A_{N1} & -D_{NN}A_{N2} & \cdots & 0 \end{bmatrix} \end{aligned}$$

将第  $i$  个智能体  $k$  时刻的纳什最优解首项作为即时控制律  $\Delta u_i(k)$ , 得到第  $i$  个

智能体的实际控制量  $u_i(k) = u_i(k-1) + \Delta u_i(k)$  作用于第  $i$  个智能体；

在下一时刻，重复上述过程求解第  $i$  个智能体的即时控制律  $\Delta u_i(k+1)$ ，进而得到整个系统的最优控制律  $\Delta u(k+1)$ ，并依次循环。

### 5.3 仿真研究

针对一个三输入三输出的分数阶多变量系统，利用本章所提的分布式动态矩阵控制方法对其进行控制，并通过 MATLAB 验证所提算法的控制性能。

$$G(s) = \begin{bmatrix} \frac{e^{-4s}}{79s^{0.95} + 1} & \frac{e^{-6s}}{87s^{0.97} + 1} & \frac{e^{-4s}}{120s^{0.92} + 1} \\ \frac{-1.25e^{-2s}}{33s^{0.90} + 1} & \frac{3.75e^{-6s}}{36s^{0.92} + 1} & \frac{e^{-3s}}{38s^{0.95} + 1} \\ \frac{-2e^{-2s}}{120s^{0.92} + 1} & \frac{2e^{-4s}}{145s^{0.95} + 1} & \frac{3.5e^{-2s}}{85s^{0.97} + 1} \end{bmatrix} \quad (5.16)$$

依据分布式预测控制的思想，将上式中的系统分解为如下形式的三个智能体子系统，并分别对各智能体使用基于分数阶系统的 PID 型动态矩阵控制。

$$\text{子系统 1:} \quad G_1(s) = \frac{e^{-4s}}{79s^{0.95} + 1} \quad (5.17a)$$

$$\text{子系统 2:} \quad G_2(s) = \frac{3.75e^{-6s}}{36s^{0.92} + 1} \quad (5.17b)$$

$$\text{子系统 3:} \quad G_3(s) = \frac{3.5e^{-2s}}{85s^{0.97} + 1} \quad (5.17c)$$

为了便于对比，将设置如下的仿真参数：在  $k=0$  时刻将设定值设置为 1；在  $k=500$  时刻，给系统施加一个振幅为 -0.2 的输出干扰；将系统两轮迭代的误差精度设置为 0.01，其他具体的参数设置详情已在表 5.1 中列出。

表 5.1 FO-DDMC 以及 DDMC 方法的参数表

Parameters	FO-DDMC	DDMC
$P$	10	
$M$	3	
$N$	300	
$T_s$	2s	
$\alpha$	1	
$\beta$	0.95	
$q_1, q_2, q_3$	\	1, 1, 1
$r_1, r_2, r_3$	0.6, 0.1, 0.5	
$k_p^1, k_p^2, k_p^3$	0.5, 0.5, 0.5	\
$k_i^1, k_i^2, k_i^3$	0.09, 0.12, 0.1	\
$k_d^1, k_d^2, k_d^3$	5.6, 5.8, 5.6	\

## (1)模型匹配

图 5.1 分别给出了模型匹配情况下分数阶分布式动态矩阵控制方法(FO-DDMC)与整数阶分布式动态矩阵控制方法(DDMC)控制下的三个智能体子系统的闭环响应对比曲线。从图 5.1a 和图 5.1b 中可以看出, FO-DDMC 方法控制下的子系统 1 和子系统 2 均具有较快的响应速度, 且子系统 1 对输出干扰的恢复能力更强; 图 5.1c 中的子系统 3 不仅拥有更好的跟踪性能及抗干扰性能, 且拥有更小的超调量。综合三个子系统的控制效果对比结果, 可以说明本章所提的 FO-DDMC 方法控制下的系统整体性能更优。

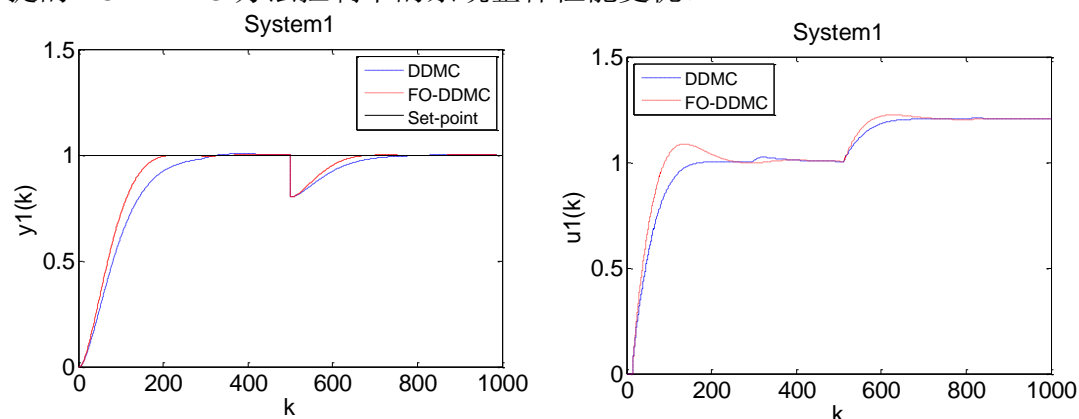


图 5.1a 模型匹配时 DDMC、FO-DDMC 方法控制下子系统 1 的闭环响应对比曲线

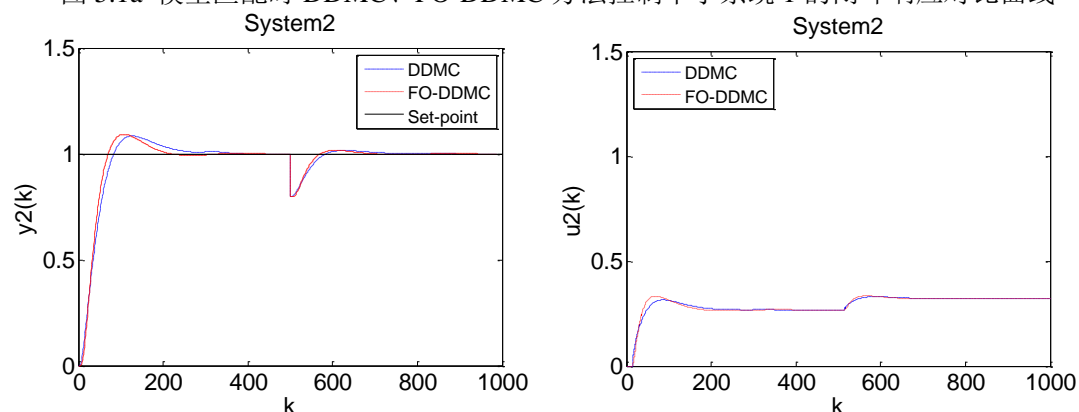


图 5.1b 模型匹配时 DDMC、FO-DDMC 方法控制下子系统 2 的闭环响应对比曲线

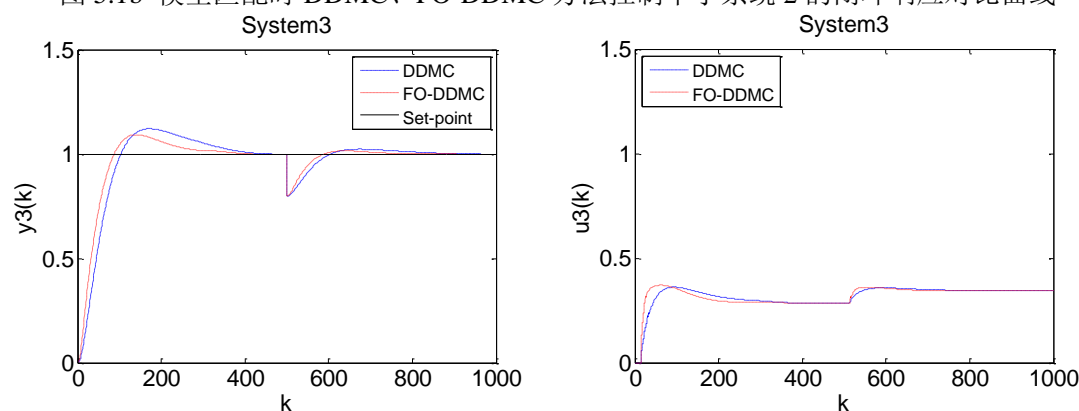


图 5.1c 模型匹配时 DDMC、FO-DDMC 方法控制下子系统 3 的闭环响应对比曲线

## (2)模型失配

由于实际建模过程中存在的各种不确定性因素往往会导致对过程建立的模



型存在一定偏差，因此有必要考虑系统在模型失配情况下的控制性能。针对本章所考虑的系统模型，通过蒙特卡洛方法可以分别产生三组失配  $\pm 30\%$  的过程参数形式如下：

第一组失配情况下的过程参数：

$$K_1 = 1.2400, T_1 = 71.3079, \alpha_1 = 0.9218, \tau_1 = 3.0669$$

$$K_2 = 3.5019, T_2 = 42.0524, \alpha_2 = 0.8468, \tau_2 = 5.6541$$

$$K_3 = 2.7271, T_3 = 64.4192, \alpha_3 = 0.9437, \tau_3 = 2.5474$$

第二组失配情况下的过程参数：

$$K_1 = 0.9912, T_1 = 100.6697, \alpha_1 = 0.9953, \tau_1 = 3.1405$$

$$K_2 = 4.6854, T_2 = 34.3100, \alpha_2 = 0.8869, \tau_2 = 7.6542$$

$$K_3 = 2.5250, T_3 = 92.9428, \alpha_3 = 0.9805, \tau_3 = 2.5208$$

第三组失配情况下的过程参数：

$$K_1 = 1.1546, T_1 = 87.4720, \alpha_1 = 0.9741, \tau_1 = 3.7413$$

$$K_2 = 3.0120, T_2 = 39.3583, \alpha_2 = 0.8743, \tau_2 = 4.3146$$

$$K_3 = 2.5470, T_3 = 73.6231, \alpha_3 = 0.8833, \tau_3 = 2.3881$$

图 5.2-图 5.4 分别给出了三组模型失配情况下 FO-DDMC 方法与 DDMC 方法控制下的三个智能体子系统的闭环响应对比曲线。由图 5.2 中可以发现，DDMC 方法控制下的子系统 1 的响应速度较慢，且抗输出干扰的恢复能力较差，FO-DDMC 方法控制下的子系统 2 和子系统 3 虽然均出现了一定超调，但在跟踪性能和抗干扰性能上仍然占有优势，且子系统 3 产生的超调更小。由图 5.3 可以看出，FO-DDMC 方法控制下的三个子系统在响应速度和抗输出干扰恢复能力上的表现均更加优越。由图 5.4 可以看出 FO-DDMC 方法控制下的三个子系统除了拥有更快的响应速度及更强的抗干扰恢复能力外，与此同时该方法控制下的子系统 2 和子系统 3 还拥有更小的超调。综上所述，对比之下，可以说明在模型失配情况下本章所提的 FO-DDMC 方法控制下的系统仍然拥有更好的整体性能。

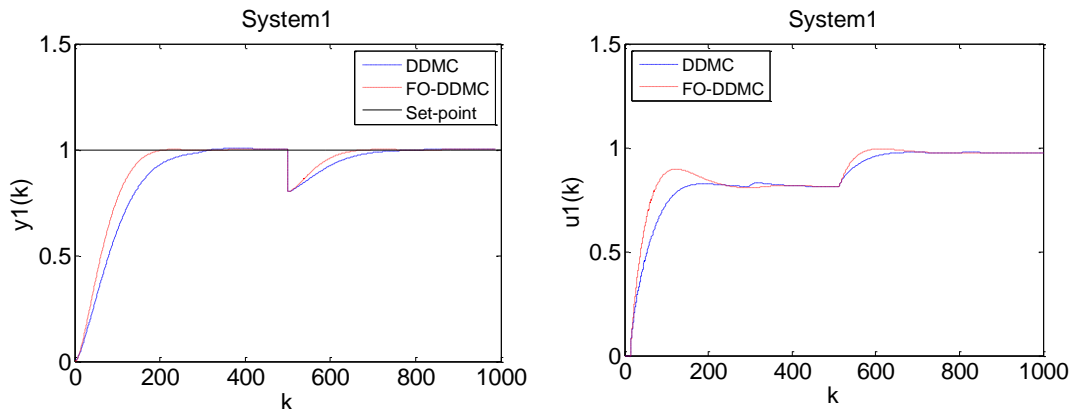


图 5.2a 第一组失配情况时 DDMC、FO-DDMC 方法控制下子系统 1 的闭环响应对比曲线

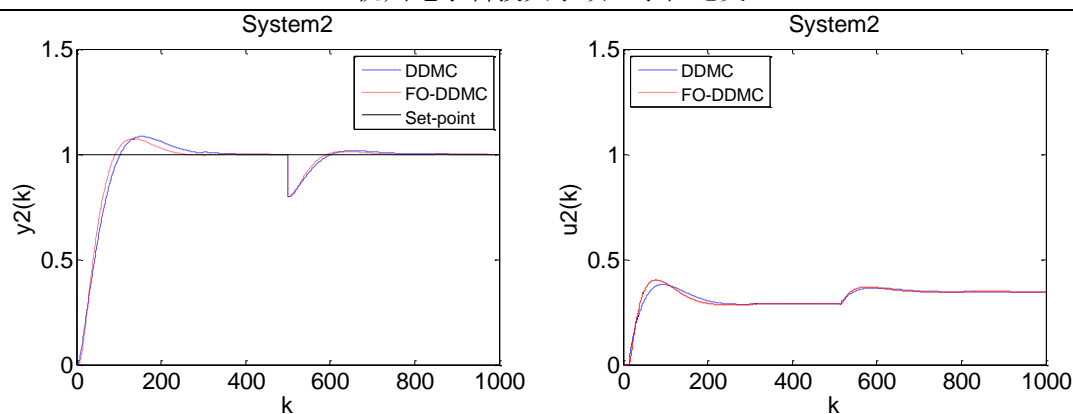


图 5.2b 第一组失配情况时 DDMC、FO-DDMC 方法控制下子系统 2 的闭环响应对比曲线

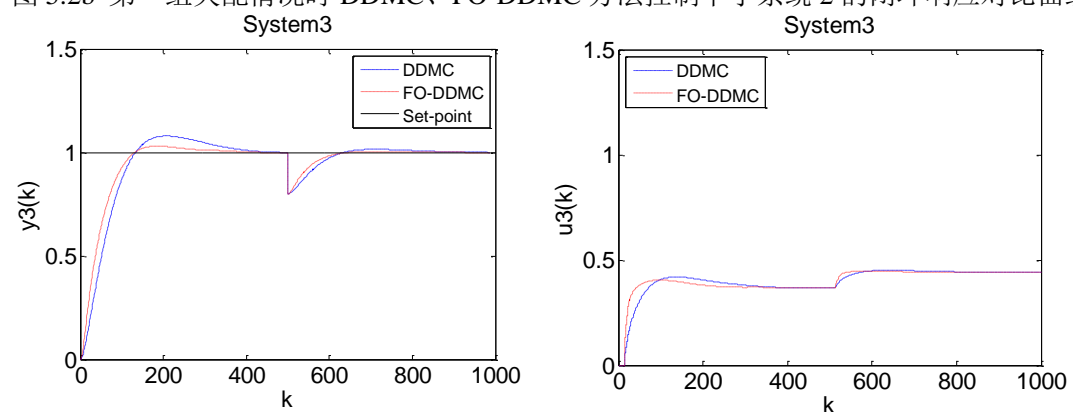


图 5.2c 第一组失配情况时 DDMC、FO-DDMC 方法控制下子系统 3 的闭环响应对比曲线

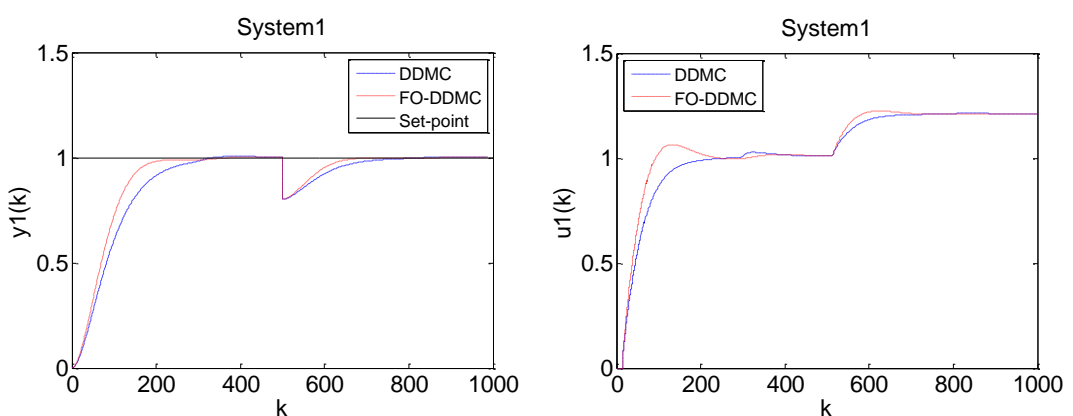


图 5.3a 第二组失配情况时 DDMC、FO-DDMC 方法控制下子系统 1 的闭环响应对比曲线

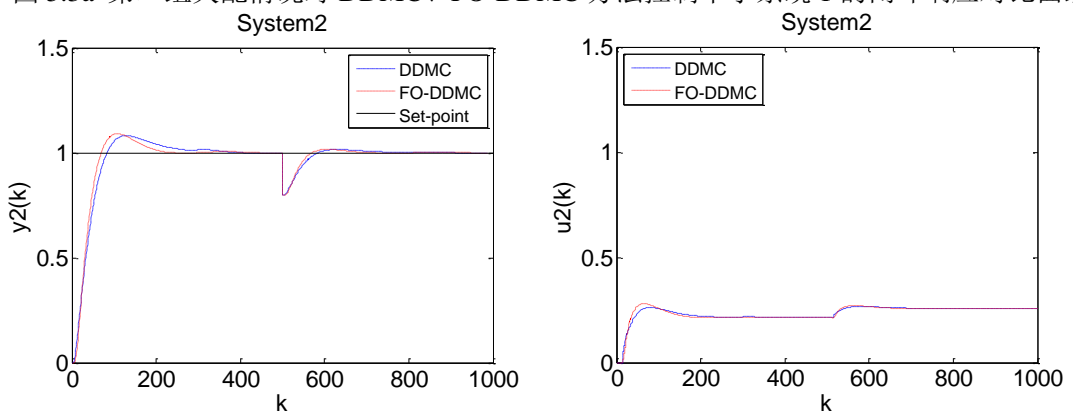


图 5.3b 第二组失配情况时 DDMC、FO-DDMC 方法控制下子系统 2 的闭环响应对比曲线

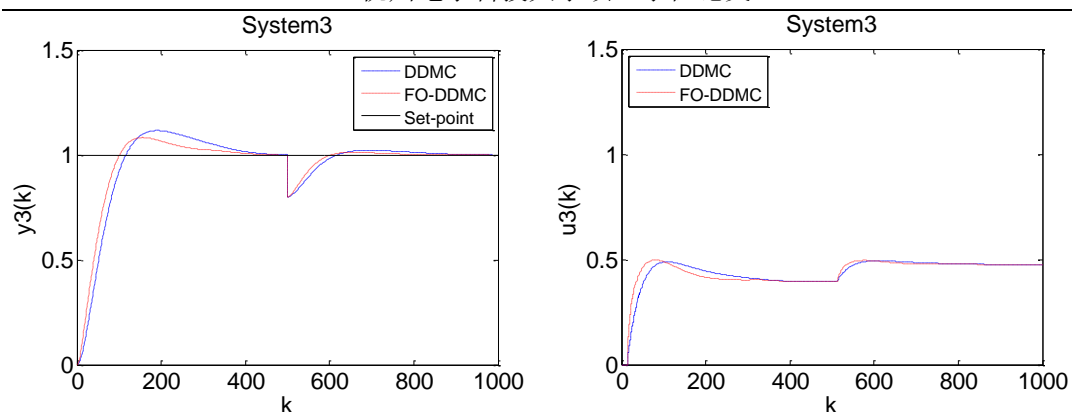


图 5.3c 第二组失配情况时 DDMC、FO-DDMC 方法控制下子系统 3 的闭环响应对比曲线

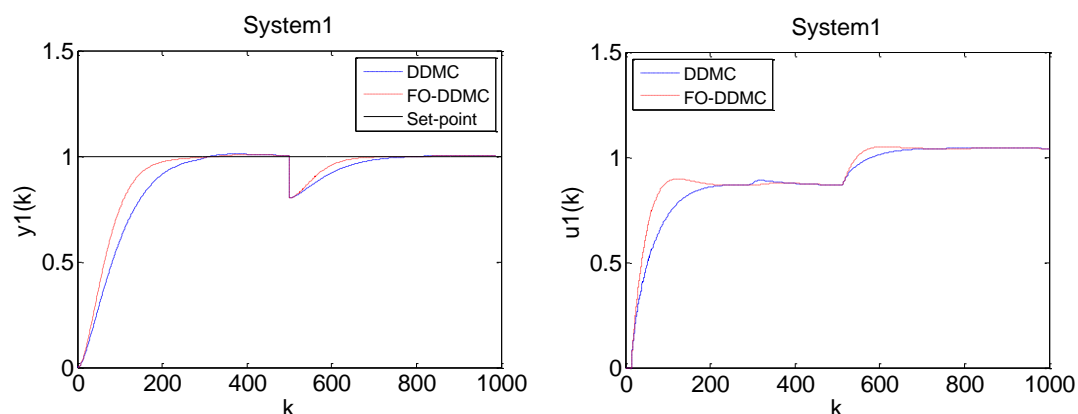


图 5.4a 第三组失配情况时 DDMC、FO-DDMC 方法控制下子系统 1 的闭环响应对比曲线

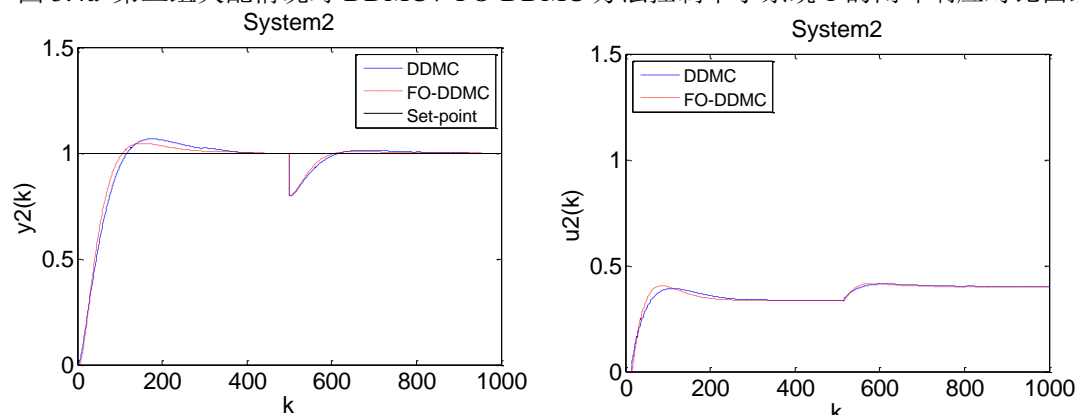


图 5.4b 第三组失配情况时 DDMC、FO-DDMC 方法控制下子系统 2 的闭环响应对比曲线

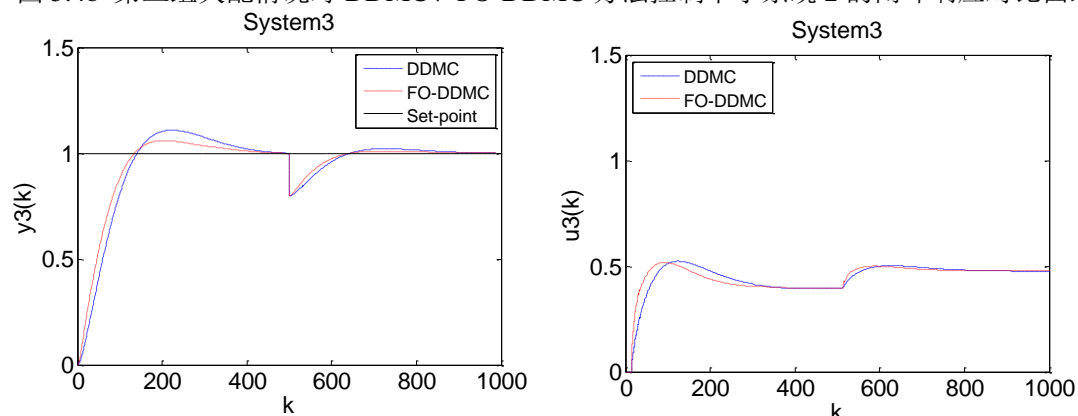


图 5.4c 第三组失配情况时 DDMC、FO-DDMC 方法控制下子系统 3 的闭环响应对比曲线

## 5.4 本章小结

本章提出了一种多变量分数阶系统的分布式动态矩阵控制方法。该方法将一类基于 PID 算子改进的分布式动态矩阵控制方法成功运用于一类多变量分数阶线性系统，在保持传统分布式动态矩阵控制优良性能的同时，也提高了控制系统参数调节的灵活性。最后通过 MATLAB 仿真对本章所提算法进行验证，仿真结果进一步说明了本章方法的有效性和可行性。

## 第 6 章 总结与展望

### 6.1 本文总结

本文主要研究了改进的动态矩阵控制算法在分数阶系统中的应用。全文主要工作包括如下三个方面：对动态矩阵控制方法的改进与对比、改进的动态矩阵控制在分数阶系统中的应用以及改进的分布式动态矩阵控制在多变量分数阶系统中的应用。

动态矩阵控制作为预测控制中的典型算法，具有对于模型要求低，鲁棒性强，处理时延简单易行等优点，在强耦合、大时滞及无法建立精确数学模型的过程中非常适用。然而该方法同时也存在在线计算量大，控制参数设计灵活性较差等局限，从而使得其在模型阶次，环境扰动及非线性等方面存在不确定性较大的控制系统中往往难以达到预期的控制效果。分数阶次的微分方程由于具有传统整数阶次微分方程所不具备的任意阶次和记忆特性，因而可以更精确的描述实际过程中的系统内在特性，这也是研究分数阶系统的意义所在。因此本文讨论了传统动态矩阵控制的改进方法，并将其进一步推广至分数阶系统的控制中。

本文主要基于单变量和多变量的分数阶系统，分别提出了改进的动态矩阵控制方法及改进的分布式动态矩阵控制方法，两种方法均通过 Oustaloup 近似方法获得分数阶系统的整数阶近似模型，然后分别使用改进的动态矩阵控制及改进的分布式动态矩阵控制方法对其进行控制。本文的研究工作主要体现在后三章，通过第三章首先介绍了两种改进的动态矩阵控制方法，并在仿真实例中与传统的 DMC 方法进行对比，结果说明了 PID 算子改进的动态矩阵控制方法拥有更好的控制性能。然后在第四章中针对一类单变量分数阶系统设计了一种分数阶系统的 PID 型动态矩阵控制器，最后在第五章针对一类多变量分数阶系统，将改进的动态矩阵控制方法运用于分布式控制结构中，设计了一种分数阶系统的分布式动态矩阵控制器。

### 6.2 研究展望

本文通过将改进的动态矩阵控制方法运用于一类分数阶系统中，在改善系统的控制性能的同时，有效推广了预测控制在分数阶系统中的应用。虽然笔者通过本文的工作取得了一定的研究成果，但鉴于本人能力有限，仍然存在一些需要进一步改进和完善的地方：

#### (1) 分数阶系统的模型建立

建模是对实际工业过程实施控制的前提，在过程控制中的地位极其重要。然而如何建立过程对象的分数阶模型至今仍然没有统一有效的公理公式。因此，有必要对分数阶的建模方法做进一步研究，以便于日后更好的研究分数阶系统中的控制问题。

#### （2）分数阶系统的时滞处理

实际工业过程中对象的时滞特性普遍存在，因而在分数阶系统的建模过程中对滞后时间的选取及处理也是需要进一步考虑的问题。

#### （3）控制参数的优化

改进的动态矩阵控制在改善系统控制性能的同时，也增加了控制参数调节的灵活性。但在控制参数的调节上多为经验整定，缺乏行之有效的选取方法。因此有必要研究一种可以对控制参数进行优化的调整方法。

#### （4）分数阶系统的稳定性

稳定性对于评价一个系统控制品质的好坏起到关键作用。在传统的整数阶控制系统中用于分析系统稳定性的方法已经相当成熟，而在分数阶控制系统中对于稳定性的分析仍然缺乏一套完备的理论体系。因此研究整数阶控制系统稳定性理论在分数阶系统中的扩展十分必要。

## 致谢

时光飞逝，岁月如梭，我在就读研究生的两年半里，目送了两届师兄师姐的毕业离校，但转瞬之间，我竟然也到了要毕业的时候。读研的两年半也许并不算长，但是人的一生能有多少两年半呢？在读研这段期间我周围的同学聪慧机敏、技艺超群；往来的老师和蔼可亲、德艺双馨；生活的环境绿树成荫、人杰地灵……我很知足也很庆幸，也许，来到这里读研是上天对我最大的眷顾吧！

首先，我要感谢我的导师张日东老师。他不但教授了我专业理论知识，给我提供了他能提供的一切实验条件，更是教授了我许多为人处世之法。想要在这个社会很好的生活下去，不止需要过硬的专业知识做支撑，更需要很好的社会经验和待人处世的技巧。我很高兴，这两年来，我学到了很多。

同时，还要感谢邹洪波老师的悉心指导，是他不厌其烦地指导着我们加热炉实验的整个过程。从实验设备的选型、搭建和操作各个环节，都离不开邹老师的耐心指导和帮助，是邹老师指导着我们如何将理论研究运用于实际过程，在这里向您道一声感谢。

其次，我要感谢我的师兄吴胜、李海生，师姐邹琴，正是在他们的鼎力帮助下，我的课题才有了很大的进展，并且每每学术上遇到瓶颈的时候，都是他们孜孜不倦的悉心指导才让我一次次克服困难。每次和他们的交流都是如沐春风般畅快、春燕回巢般舒适，让我受益匪浅。

再次，我要感谢我的同学王成龙、王玉中、徐卫德、戴宇丰、唐飞平、高腾龙。很高兴也很荣幸的和他们成为一个实验室的同学，刘禹锡曾说过他的朋友圈是“谈笑有鸿儒，往来无白丁”。我想这两年半我在实验室的同学圈也是这样的，他们谈吐儒雅、不矜不伐，正所谓近朱者赤，和他们在一起的时光里使我自身得到了很大的提升。

另外，我还要感谢同实验室的师弟师妹们，在实验室的学习生活中，我得到了很多无微不至的帮助，这让我感受到了实验室温馨的一面。

最后，我要感谢杭州电子科技大学自动化学院的所有老师，在我读研的两年半期间，帮助我解决问题，克服问题，得以让我能顺利毕业。

正所谓“大恩不言谢”，“滴水之恩，当以涌泉相报”，知恩图报是中华民族的传统美德。我在离开学校进入社会后一定坚守“笃学力行、守正求新”的校训，努力工作，绝不给母校抹黑，同时绝不辜负我的学校、学院、老师以及所有帮助过我的同学们！

2017 年 3 月于杭州

## 参考文献

- [1] Propoi A I. Use of linear programming methods for synthesizing sampled-data automatic systems[J]. Automation & Remote Control, 1963, 24(7): 837-844.
- [2] Richalet J, Rault A, Testud J L, et al. Model predictive heuristic control: Applications to industrial processes[J]. Automatica, 1978, 14(5): 413-428.
- [3] Garcia C E, Prett D M, Morari M. Model predictive control: theory and practice—a survey[J]. Automatica, 1989, 25(3): 335-348.
- [4] Qin S J, Badgwell T A. A survey of industrial model predictive control technology[J]. Control Engineering Practice, 2003, 11(7): 733-764.
- [5] Chen Y Q, Moore K L. Analytical stability bound for a class of delayed fractional-order dynamic systems[J]. Nonlinear Dynamics, 2002, 29(1): 191-200.
- [6] Maione G. A Rational Discrete Approximation to the Operator  $s^{0.5}$ [J]. IEEE Signal Processing Letters, 2006, 13: 141-144.
- [7] Machado J A T. Analysis and design of fractional-order digital control systems[J]. Systems Analysis Modelling Simulation, 1997, 27(2-3): 107-122.
- [8] Vinagre B M, Podlubny I, Hernández A, et al. Some approximations of fractional order operators used in control theory and applications[J]. Fractional Calculus & Applied Analysis, 2000, 3(3): 231-248.
- [9] Al-Alaoui M A. Filling the gap between the bilinear and the backward-difference transforms: An interactive design approach[J]. International Journal of Electrical Engineering Education, 1997, 34(4): 331-337.
- [10] Al-Alaoui M A. A class of second-order integrators and low-pass differentiators[J]. IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, 1995, 42(4): 220-223.
- [11] Chen Y Q, Vinagre B M, Podlubny I. Continued fraction expansion approaches to discretizing fractional order derivatives—an expository review[J]. Nonlinear Dynamics, 2004, 38(1-4): 155-170.
- [12] Chen Y Q, Vinagre B M, Podlubny I. A new discretization method for fractional order differentiators via continued fraction expansion[C]. ASME 2003 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. American Society of Mechanical Engineers, 2003: 761-769.
- [13] Oustaloup A, Levron F, Mathieu B, et al. Frequency-band complex noninteger



- differentiator: characterization and synthesis[J]. IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, 2000, 47(1): 25-39.
- [14] 赵春娜. 分数阶系统分析与设计方法研究[D]. 东北大学, 2006.
- [15] Oustaloup A, Bansard M. First generation CRONE control[C]. Systems, Man and Cybernetics, 1993.'Systems Engineering in the Service of Humans', Conference Proceedings., International Conference on. IEEE, 1993: 130-135.
- [16] Oustaloup A, Mathieu B, Lanusse P. Second generation CRONE control[C]. Systems, Man and Cybernetics, 1993.'Systems Engineering in the Service of Humans', Conference Proceedings., International Conference on. IEEE, 1993: 136-142.
- [17] Lanusse P, Oustaloup A, Mathieu B. Third generation CRONE control[C]. Systems, Man and Cybernetics, 1993.'Systems Engineering in the Service of Humans', Conference Proceedings., International Conference on. IEEE, 1993: 149-155.
- [18] Oustaloup A, Bluteau B, Nouillant M. First generation scalar CRONE control: application to a two DOF manipulator and comparison with non linear decoupling control[C]. Systems, Man and Cybernetics, 1993.'Systems Engineering in the Service of Humans', Conference Proceedings., International Conference on. IEEE, 1993: 453-458.
- [19] Yousfi N, Melchior P, Lanusse P, et al. Decentralized CRONE control of nonsquare multivariable systems in path-tracking design[J]. Nonlinear Dynamics, 2014, 76(1): 447-457.
- [20] Sabatier J, Poullain S, Latteux P, et al. Robust speed control of a low damped electromechanical system based on CRONE control: application to a four mass experimental test bench[J]. Nonlinear Dynamics, 2004, 38(1-4): 383-400.
- [21] Cervera J, Baños A. Automatic loop shaping in QFT by using CRONE structures[J]. IFAC Proceedings Volumes, 2010, 39(11): 207-212.
- [22] Cervera J, Baños A. Automatic loop shaping in QFT using CRONE structures[J]. Journal of Vibration and Control, 2008, 14(9-10): 1513-1529.
- [23] Altet O, Moreau X, Moze M, et al. Principles and synthesis of hydractive CRONE suspension[J]. Nonlinear Dynamics, 2004, 38(1-4): 435-459.
- [24] Podlubny I. Fractional-order systems and  $PI^{\lambda}D^{\mu}$  controllers [J]. IEEE Transactions on Automatic Control, 1999, 44(1): 208-214.
- [25] 薛定宇, 陈阳泉. 控制数学问题的matlab求解[M].2009.

- [26] 严慧. 分数阶 $PI^\lambda D^\mu$ 控制器的设计及数字实现[D]. 南京航空航天大学, 2007: 41-51.
- [27] 梁涛年. 分数阶PID控制器及参数不确定分数阶系统稳定域分析[D]. 西安电子科技大学, 2011.
- [28] Monje C A, Calderon A J, Vinagre B M, et al. On fractional  $PI^\lambda$  controllers: some tuning rules for robustness to plant uncertainties[J]. *Nonlinear Dynamics*, 2004, 38(1-4): 369-381.
- [29] Valério D, da Costa J S. Tuning of fractional PID controllers with Ziegler–Nichols-type rules[J]. *Signal Processing*, 2006, 86(10): 2771-2784.
- [30] Rahimian M A, Tavazoei M S. Stabilizing fractional-order PI and PD controllers: an integer-order implemented system approach[J]. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 2010, 224(8): 893-903.
- [31] Bouafoura M K, Braiek N B.  $PI^\lambda D^\mu$  controller design for integer and fractional plants using piecewise orthogonal functions[J]. *Communications in Nonlinear Science and Numerical Simulation*, 2010, 15(5): 1267-1278.
- [32] Richalet J. Industrial applications of model based predictive control[J]. *Automatica*, 1993, 29(5): 1251-1274.
- [33] Rodrigues M A, Odloak D. An infinite horizon model predictive control for stable and integrating processes[J]. *Computers & Chemical Engineering*, 2003, 27(8-9): 1113-1128.
- [34] Zhang R D, Xue A K, Wang S Q. Dynamic Modeling and Nonlinear Predictive Control Based on Partitioned Model and Nonlinear Optimization[J]. *Industrial & Engineering Chemistry Research*, 2011, 50(13): 8110-8121.
- [35] Zhang R D, Gao F R. Multivariable decoupling predictive functional control with non-zero-pole cancellation and state weighting: Application on chamber pressure in a coke furnace[J]. *Chemical Engineering Science*, 2013, 94: 30-43.
- [36] Zhang R D, Xue A K, Wang S Q, et al. An improved model predictive control approach based on extended non-minimal state space formulation[J]. *Journal of Process Control*, 2011, 21(8): 1183-1192.
- [37] Beschi M, Berenguel M, Visioli A, et al. Implementation of feedback linearization GPC control for a solar furnace[J]. *Journal of Process Control*, 2013, 23(10): 1545-1554.
- [38] Marchetti A G, Ferramosca A, González A H. Steady-state target optimization

- designs for integrating real-time optimization and model predictive control[J]. Journal of Process Control, 2014, 24(1): 129-145.
- [39] Skrjanc I, Matko D. Predictive functional control based on fuzzy model for heat-exchanger pilot plant[J]. IEEE Transactions on Fuzzy Systems, 2000, 8(6): 705-712.
- [40] Oblak S, Skrjanc I. Multivariable fuzzy predictive functional control of a MIMO nonlinear system[C]. Intelligent Control, 2005. Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation. IEEE, 2005: 1029-1034.
- [41] Su C L, Li P. Adaptive predictive functional control based on Takagi-Sugeno model and its application to pH process[J]. Journal of Central South University of Technology, 2010(2): 363-371.
- [42] Li N, Li S Y, Xi Y G. Multi-model predictive control based on the Takagi-Sugeno fuzzy models: a case study[J]. Information Sciences, 2004, 165(3): 247-263.
- [43] Bououden S, Chadli M, Filali S, et al. Fuzzy model based multivariable predictive control of a variable speed wind turbine: LMI approach[J]. Renewable Energy, 2012, 37(1): 434-439.
- [44] Tsai P F, Chu J Z. Developing a robust model predictive control architecture through regional knowledge analysis of artificial neural network[J]. Journal of Process Control, 2003, 13(5): 423-435.
- [45] 薛福珍, 柏洁. 基于先验知识和神经网络的非线性建模与预测控制[J]. 系统仿真学报, 2004, 16(5): 1057-1059.
- [46] 石屹然, 田彦涛, 史红伟. 基于Modified Volterra模型的SI发动机空燃比非线性模型预测控制[J]. 吉林大学学报: 工学版, 2014, 44(2): 538-547.
- [47] 黄德先, 金以慧, 张杰. 基于小波网络的非线性多变量约束预测控制[J]. 清华大学学报, 2000, 40(9): 116-119.
- [48] Feng K, Lu J, Chen J. Nonlinear model predictive control base on support vector machine and genetic algorithm [J]. Chinese Journal of Chemical Engineering, 2015, 23(12): 2048-2052.
- [49] 张日东, 王树青. 基于支持向量机的一类非线性系统预测控制[J]. 控制与决策, 2007, 22(10): 1103-1107.
- [50] 张健中. 一类连续化工生产过程的模型辨识及非线性预测控制研究[D]. 哈尔滨工业大学, 2010.

- [51] 李妍, 毛志忠, 王琰, 等. 基于多面体终端域的 Hammerstein-Wiener 非线性预测控制[J]. 自动化学报, 2011, 37(5): 629-638.
- [52] 史芸, 田学民. 一种基于 Wiener 模型的非线性预测控制[C]. 中国过程控制会议. 2006: 387-390.
- [53] 邹志云, 郭宇晴, 王志甄等. 非线性 Hammerstein 模型预测控制策略及其在 pH 中和过程中的应用[J]. 化工学报, 2012, 63(12): 3965-3970.
- [54] 朱齐丹, 冷欣. 基于约束预测控制的增压锅炉汽包水位系统研究[J]. 哈尔滨工程大学学报(英文版), 2009, 30(11): 1246-1250.
- [55] 王喜庆, 冯晓露. 电厂过热蒸汽温度的有约束预测控制[J]. 动力工程学报, 2010, 30(3): 192-195.
- [56] 于洋, 许鋈, 罗雄麟. 预测控制约束边界效应与解决方法研究[J]. 自动化学报, 2014, 40(9): 1922-1932.
- [57] 张晓颖, 李德伟, 席裕庚等. 六自由度机械臂约束预测控制系统的设计[J]. 控制理论与应用, 2014, 31(11): 1464-1472.
- [58] Bordons C, Arce A, Del Real A J. Constrained predictive control strategies for PEM fuel cells[C]. 2006 American Control Conference. IEEE, 2006: 6 pp.
- [59] Mayne D Q, Raković S V, Findeisen R, et al. Robust output feedback model predictive control of constrained linear systems: Time varying case[J]. Automatica, 2009, 45(9): 2082-2087.
- [60] Perez T, Goodwin G C. Constrained predictive control of ship fin stabilizers to prevent dynamic stall[J]. Control Engineering Practice, 2008, 16(4): 482-494.
- [61] 刘金琨, 尔联洁. 多智能体技术应用综述[J]. 控制与决策, 2001, 16(2): 133-140.
- [62] Kayacan E, Kayacan E, Ramon H, et al. Distributed nonlinear model predictive control of an autonomous tractor-trailer system[J]. Mechatronics, 2014, 24(8): 926-933.
- [63] Stewart B T, Wright S J, Rawlings J B. Cooperative distributed model predictive control for nonlinear systems[J]. Journal of Process Control, 2011, 21(5): 698-704.
- [64] Li H, Shi Y, Yan W. On neighbor information utilization in distributed receding horizon control for consensus-seeking[J]. IEEE transactions on cybernetics, 2016, 46(9): 2019-2027.
- [65] Farina M, Scattolini R. Distributed predictive control: a non-cooperative algorithm with neighbor-to-neighbor communication for linear systems[J].

- Automatica, 2012, 48(6): 1088-1096.
- [66] Perizzato A, Farina M, Scattolini R. Formation control and collision avoidance of unicycle robots with distributed predictive control[J]. IFAC-PapersOnLine, 2015, 48(23): 260-265.
- [67] Gao Y, Xia Y, Dai L. Cooperative distributed model predictive control of multiple coupled linear systems[J]. IET Control Theory & Applications, 2015, 9(17): 2561-2567.
- [68] 董海, 王宛山, 李彦平等. 分布式MPC在网络化制造环境下SCM中的应用[J]. 系统仿真学报, 2007, 19(6): 1354-1357.
- [69] 张永, 常学飞, 韩春雷等. 基于分布式模型预测控制算法在AGC系统的应用研究[J]. 吉林电力, 2013, 41(5): 13-16.
- [70] 陈庆, 李少远, 席裕庚. 一类串联生产过程的分布式解耦预测控制[J]. 控制与决策, 2004, 19(6): 647-650.
- [71] 张舒展, 赵东亚, 朱全民. 基于纳什最优的分布式预测控制在常减压塔系统中的应用[J]. 计算机与应用化学, 2016, 33(3).
- [72] 李纹, 穆文英, 崔宝同. 切换非线性系统的分布式模型预测控制[J]. 控制与决策, 2014(4): 599-604.
- [73] Monje C A, Chen Y Q, Vinagre B M, et al. Fractional-order systems and controls: fundamentals and applications[M]. Springer Science & Business Media, 2010.
- [74] Podlubny I. Fractional Differential Equations [M]. San Diego: Academic Press, 1999, 244-257.
- [75] 李元凯. 分数阶控制系统理论分析方法[D]. 重庆邮电大学, 2006.
- [76] 席裕庚. 预测控制.第2版[M]. 国防工业出版社, 2013.
- [77] 窦秀华. 啤酒发酵温度SMITH补偿分布式预测控制算法研究[D]. 青岛科技大学, 2009.
- [78] Romero M, de Madrid A P, Vinagre B M. Arbitrary real-order cost functions for signals and systems[J]. Signal Processing, 2011, 91(3): 372-378.
- [79] Ren P, Chen G Y, Pei H L. A Simplified Algorithm for Dynamic Matrix Control with PID Structure[C]. Intelligent Computation Technology and Automation (ICICTA), 2008 International Conference on. IEEE, 2008, 1: 279-283.
- [80] 詹雪兰. 催化裂化主分馏塔和吸收稳定系统的全流程模拟与优化[D]. 华东理工大学, 2014.

## 附录

### 发表的学术论文:

- [1] Dawei Wang, Junfeng Zhang, Ridong Zhang, Furong Gao. A new design of fractional order dynamic matrix control with PID-type structure. (ISA TRANSACTIONS, 已投, SCI, IF=2.6)
- [2] Dawei Wang, Ridong Zhang, Furong Gao. Design of distributed PID type dynamic matrix controller for fractional order systems. (Optimal Control, Applications and Methods, 已投, SCI, IF=1.097)
- [3] Dawei Wang, Jili Tao, Ridong Zhang. Design of a PID-type Fractional Dynamic Matrix control. (已投 The 36th Chinese Control Conference 2017)

### 公开的发明专利:

- [1] 汪大卫, 张日东, 张俊锋. 一种工业加热炉系统的分数阶动态矩阵控制方法. 发明专利, 公开号: CN 105892296 A.
- [2] 汪大卫, 张日东. 一种批次注塑过程的滚动时域跟踪控制方法. 发明专利, 公开号: CN 105353619 A.
- [3] 张日东, 汪大卫. 一种焦炭炉炉膛压力的分布式 PID 型动态矩阵控制方法. 发明专利, 公开号: CN 106444388 A.
- [4] 张日东, 汪大卫. 一种废塑料裂解炉炉膛温度的分布式 PID 型预测函数控制方法. 发明专利, 公开号: CN 106444362 A.
- [5] 张俊锋, 汪大卫, 张日东. 一种工业加热炉温度系统的稳定切换控制器设计方法. 发明专利, 公开号: CN 105911884 A.
- [6] 张日东, 汪大卫, 张俊锋. 分布式动态矩阵控制优化的焦炭炉炉膛压力控制方法. 发明专利, 公开号: CN 105955014 A.
- [7] 张日东, 陶吉利, 汪大卫. 加热炉氧含量的 GA 优化 T-S 模糊 ARX 建模方法. 发明专利, 公开号: CN 105334730 A.
- [8] 张日东, 邹洪波, 汪大卫. 一种批次注塑过程的稳定控制器设计方法. 发明专利, 公开号: CN 105334751 A.
- [9] 张日东, 张俊锋, 汪大卫. 一种非自衡对象的分布式动态矩阵控制方法. 发明专利, 公开号: CN 106200379 A.

### 已受理的发明专利:

- [1] 汪大卫, 张日东. 炼油加热炉炉膛压力的分数阶分布式动态矩阵控制方法. 发明专利, 申请号: 201611252446.9.