



中国热管理网

WWW.REGUANLI.COM

分类号\_\_\_\_\_

学校代码 10487

学号 M200871538

密级\_\_\_\_\_

# 华中科技大学

# 硕士学位论文

## 固体氧化物燃料电池热管理系统的 研究与设计

学位申请人 马腾

学科专业：控制理论与控制工程

指导教师：李曦 副教授

答辩日期：2011年2月23日

**A Thesis Submitted in Partial Fulfillment of the Requirements  
for the Degree of Master of Engineering**

**The Research and Design of SOFC Thermal  
Management System**

**Candidate : Ma Teng**

**Major : Control Theory and Control Engineering**

**Supervisor : Ass. Prof. Li Xi**

**Huazhong University of Science & Technology**

**Wuhan 430074, P.R.China**

**February, 2011**

## 独创性声明

本人声明所呈交的学位论文是我个人在导师的指导下进行的研究工作及取得的研究成果。尽我所知，除文中已标明引用的内容外，本论文不包含任何其他人或集体已经发表或撰写过的研究成果。对本文的研究做出贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：

日期： 年 月 日

## 学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，即：学校有权保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权华中科技大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

保密，在\_\_\_\_\_年解密后适用本授权数。

本论文属于

不保密.

(请在以上方框内打“√”)

学位论文作者签名：

指导教师签名：

日期： 年 月 日

日期： 年 月 日

## 摘 要

固体氧化物燃料电池 (SOFC) 具有高能量转化效率、零污染、零噪声等优点, 因此是一种十分理想的可替代能源。SOFC 在应用中, 要保持安全、高效的运行状态, SOFC 电堆的工作温度环境起着关键作用, 为此必须为其设计出合理的 SOFC 热管理系统对其工作温度进行合理、有效的控制。SOFC 热管理系统的设计需要考虑的因素很多, 主要包括工艺结构的设计、控制策略的设计、控制策略的实现等几个重要方面, 本文将从以上这三个方面入手对 SOFC 热管理系统进行设计。

本文的工作主要包括: 首先, 从工艺结构上对 SOFC 热管理系统进行详细分析, 并结合试验数据采用 T-S 模糊建模方法建立 SOFC 热管理系统的模型; 然后, 基于建立的模型, 采用普通 PI 控制方式和串级 PI 控制方式对 SOFC 的工作温度进行控制, 通过测试发现串级 PI 控制能更有效的控制 SOFC 的工作温度, 从而完成了控制策略的设计; 接着, 采用 XPC 快速控制原型方式将仿真下的控制策略转化为 XPC 平台下的实时控制代码, 对控制策略进行验证, 完成了控制策略的实现; 最后, 为了从体积庞大的 XPC 平台顺利过渡到小巧的 MPC555 平台, 以提高系统的集成度向产品级目标迈进, 完成了 UCOS 实时操作系统在 MPC555 平台上的移植。

**关键词:** SOFC 热管理系统 T-S 模糊模型 串级 PI 控制 XPC 快速控制原型 UCOS 实时操作系统

## Abstract

SOFC have the advantages of high energy conversion, zero pollution and zero noise, and is a kind of perfect alternative energy. In the application of SOFC, in order to keep it in a safe and high efficient running state, its working temperature plays a key role, so it is necessary to design a reasonable and efficient thermal management system to control it. The design for SOFC thermal management system need to think about a lot of elements including the design for process structures, the design for control strategy and the realization of the control strategy, and in this paper they will be discussed to complete the design for SOFC thermal management system.

The main work of this thesis are as follows: firstly, analyzing SOFC thermal management system carefully from the process structures aspect, then based on the experimental data, adopting T-S fuzzy model created method to create the mathematical model of SOFC thermal management system; secendly, based on the mathematical model adopting PI control strategy and cascade PI control strategy to control the working temperatruue of SOFC respectively, finding that cascade PI control strategy has better control effect; thirdly, using rapid control prototyping method of XPC to convert the control strategy of the former into real-time control code of the XPC platform to verify that the control strategy are effective; finally, in order to improve the integration of the system to product level, a trasion from XPC to MPC555 is needed, then the transplant of UCOS real-time operating system based on the MPC555 is accomplished.

**Key words:** SOFC thermal management system, T-S fuzzy model, cascade PI control, rapid control prototyping of XPC, UCOS real-time operating system

# 目 录

摘 要 .....	I
Abstract .....	II
<b>1 绪论</b>	
1.1 课题研究的意义 .....	(1)
1.2 国内外研究状况 .....	(2)
1.3 论文的主要工作和章节安排 .....	(6)
<b>2 热管理系统模型的搭建</b>	
2.1 SOFC 系统的工艺结构 .....	(8)
2.2 建模方法简介及选择 .....	(10)
2.3 SOFC 系统的 T-S 模糊建模 .....	(11)
2.4 本章小结 .....	(16)
<b>3 热管理系统控制策略的研究</b>	
3.1 逻辑控制过程控制器的设计 .....	(17)
3.2 连续动态过程控制器的设计 .....	(21)
3.3 逻辑控制与连续控制的统一 .....	(28)
3.4 本章小结 .....	(31)
<b>4 控制策略在 XPC 平台的实现和验证</b>	
4.1 快速控制原型与 XPC 平台 .....	(33)
4.2 XPC 平台的搭建 .....	(35)
4.3 硬件在回路仿真与控制策略验证 .....	(45)
4.4 本章小结 .....	(46)

<b>5 UCOS 实时操作系统在 MPC555 上的移植</b>	
5.1 MPC555 硬件平台简介 .....	(47)
5.2 UCOS 实时操作系统简介 .....	(49)
5.3 UCOS 操作系统的内核结构 .....	(50)
5.4 UCOS 系统的移植工作的展开 .....	(52)
5.5 本章小结.....	(55)
<b>6 总结与展望</b> .....	(57)
<b>致 谢</b> .....	(60)
<b>参考文献</b> .....	(61)
<b>附 录 攻读学位期间发表的论文</b> .....	(65)

# 1 绪论

## 1.1 课题研究的意义

随着当今社会的高速发展，人们对能源的需求量也越来越大，目前能源体系中有 80% 被煤、石油、天然气等不可再生能量牢牢占据。世界的原油价格从 2008 年起至今翻了两倍有余，还有今年的我国的柴油荒都凸显出能源短缺现象。随着社会的继续发展，能源需求量势必会越来越大，同时这些不可再生能源越来越少，能源危机将会进一步加剧。这些不可再生能源都是先将化学能转化为热能，然后再由热能转化为其它能量，转化效率低下，一般都在 33% 以下；同时这些不可再生能源产生了大量的生活污染，对人们的健康产生了很多不良影响，当今肺癌发病率变高就是大气受到污染的严重后果。并且这些不可再生能源燃烧后产生了大量的  $\text{CO}_2$ ，它是引起温室效应的主要因素。因此目前世界各国都投入了大量的资源对新能源领域进行研究，希望找到转化效率高，无污染的绿色能源；其中燃料电池就属于其中一种比较有发展潜力的新型能源。

固体氧化物燃料电池（Solid Oxide Fuel Cell, SOFC）是一种将化石燃料中的化学能直接转换为电能电化学装置，其中没有燃烧过程和机械运动，并且反应产物只有水，从而使其具有高效率、零污染、零噪声等特点<sup>[1, 2]</sup>。与其它燃料电池相比，SOFC 具有转换效率更高、燃料来源更广泛（天然气、生物质气体、采油以及其它碳氢化合物等）、无 CO 电极毒化、无液态熔融腐蚀、成本相对低（无使用 Pt 等贵金属电极材料）以及热电联供或二次发电等独特优点<sup>[3]</sup>，在固定电站、移动电源、交通运输以及军事领域有着广泛的市场前景；其成功应用对于缓解能源危机、满足电力需求、保护生态环境以及保障国家安全都有重大的意义<sup>[4]</sup>。

SOFC 主要类型有分平板型、管型和圆柱型三种，其中平板型因功率密度高和制作成本低而成为 SOFC 的发展趋势。所以，本课题以平板型中温 SOFC（电堆正常工作温度为  $750^\circ\text{C}$ ）为研究对象。SOFC 作为一种环保、高效的发电装置，实际应用最直接的

要求是 SOFC 系统能独立输出电能的同时可实时动态地跟踪外部负载。在负载动态跟踪过程中, 要维持系统高效、稳定、安全的运行, SOFC 电堆的工作温度环境起着关键的作用。电堆温度过低时, 电池片功率密度小、发电效率低下; 电堆温度过高、温差过大以及温度场分布不均时都会导致密封材料快速老化, 电池片和连接体发生形变, 甚至断裂, 从而电堆性能急剧下降。目前从课题组的材料加工工艺以及能量转化效率和安全性角度进行综合考虑, 本课题中 SOFC 最佳工作温度应该在 750℃。由于 SOFC 的电化学反应是放热反应, 并且放出的热量与负载电流相关, 如果不及时将热量散出, SOFC 会因为温度过高而损坏。在实验室研究中, 为了让 SOFC 工作在最佳工作温度上, 通常的做法是通过一个高温电炉来保障它的工作温度恒定。由于高温电炉不仅耗电量巨大, 并且体积庞大, 不可能和 SOFC 一起组成独立的发电系统。为了让 SOFC 走下试验台运用于实际成为独立的发电系统, 最大的挑战在于如何维持 SOFC 的工作温度始终处于最佳工作温度上, 以便让其安全、高效的运行。为了达到这个目的就必须配备外围热管理系统对 SOFC 的工作温度进行有效、合理的控制, 使其始终处于最佳工作温度上。

目前, 国内外关于 SOFC 技术的研究文献和专利逐年上升, 其研究的内容覆盖了 SOFC 技术的多个方面, 包括电极和电解质材料及其制备、单电池设计与制备、电堆结构设计、连接体和密封材料及其制备等, 各技术方面的发展相对均衡<sup>[5, 6]</sup>, 但在电堆和系统的控制技术相对缺乏, 进行 SOFC 电堆集成控制技术的研究迫在眉睫, 因此对 SOFC 外围热管理系统进行研究, 设计出满足要求的热管理系统实现对 SOFC 工作温度的合理控制让其安全、高效的运行具有十分重大的意义。

## 1.2 国内外研究状况

SOFC 外围热管理系统的研究是一项复杂的工程, 它必须对工艺结构、控制策略、控制策略的实现等因素进行综合考虑。

目前, 国外 SOFC 系统的研究开发主要集中在 SOFC 专门研究机构和公司, 如美国 SECA 计划资助的 FuelCell Energy (Versa Power)、Delphi、Cummins、Acumentrics、Siemens -Westinghouse 等公司以及欧洲、日本和澳大利亚的公司和研究机构。在中温

平板 SOFC 领域, 美国的 Versa Power (原加拿大 Global Thermoelectric) 领先世界水平, 早在 8 年前已经展示了第一代 2kW 中温平板型 SOFC 发电系统, 最近又推出了 3~10kW 的中温 SOFC 系统<sup>[7]</sup>; 同时, Versa Power 正在 IGFC 计划支持下开展更大功率电堆和系统的研究, 为百兆瓦级 SOFC 电站开展前期研究工作。日本三菱重工与中部电力合作, 共同开发板型 SOFC。2009 年, 三菱重工宣布其 200kW 级 SOFC-MGT (Micro Gas Turbine) 混合发电系统已成功运行 3000h<sup>[8]</sup>。日本京瓷和 Toto 合作正在商业化千瓦级的 SOFC 发电系统。澳大利亚的 CFCL (Ceramic Fuel Cell Limited) 目前正在商业化其 1kW 级的 SOFC 独立发电系统。总的说来, 世界各国都还在致力于 1~10kW 独立 SOFC 发电系统的研究和开发, 运行时间基本在 10,000h 以内; 个别公司如三菱重工 (MHI) 开展了高温工作的百 kW 级 SOFC 系统的研发工作, 旨在与 MGT 联用形成 SOFC-MGT 混合发电系统。最近, 位于美国加州 Sunnyvale 的 Bloom Energy 公司接受了美国 CBS 公司的专访, 宣布向市场推出 100kW 级 SOFC 独立发电系统, 在国际上引起了极大的关注。目前, Bloom Energy 正以每天生产 100kW 的进度, 加快 SOFC 的产业化进程<sup>[9]</sup>。但是, 上述这些公司的很多研究成果都属于商业机密, 并未在报刊或杂志上公开发表。

我国最早关于 SOFC 的研究始于 1970 年代。1990 年以后, 在国家科技部, 国家计委 (现国家发展与改革委员会)、中国科学院、教育部所属高校、地方政府等的资助下, 华中科技大学、中科院上海硅酸盐所、中科院大连化物所、中科院宁波材料研究所、中国科学院过程工程研究所 (原化冶所)、中国科学技术大学、中国矿业大学、上海交通大学、吉林大学、西安交通大学、哈尔滨工业大学、华南理工大学、中科院山西煤炭所等多家研究机构对在 SOFC 关键材料和制备工艺以及系统集成等方面相继开展了探索和研究工作, 积累了丰富的经验<sup>[10, 11]</sup>。中科院宁波材料所王蔚国研究员领导的团队已设计完成系统工艺、实物装配、燃料热处理等设计和研制工作, 并搭建了系统管路平台, 近两年, 中科院大化所程谟杰研究员领导的团队和上海硅酸盐研究所王绍荣研究员领导的团队也在逐步开展 SOFC 系统的相关研究, 其他国内单位相关研究工作正在推进。但是, 由于起步较晚和资金投入较少, 我国 SOFC 研发的总体水平与美国、德国、日本等发达国家的先进水平存在着差距, 尤其是在电堆性能与系统集成与控制方面差距较大。

在充分参考了国内外可以查阅的 SOFC 外围热管理系统文献的基础上, 本文对相关研究成果及其存在的不足做出了如下概括和总结:

在 SOFC 外围热管理系统的工艺结构设计上, 目前得到了不少积累, 如: 王礼进和张会生等人从系统、节能、提高发电效率的角度出发, 在仿真模拟的环境下, 利用电堆尾气的热量对电堆反应气体初步预热, 然后调节电堆入口空气流量带走电堆电化学反应所产生的热量, 从而控制电堆温度<sup>[12, 13]</sup>。以上热管理思路是通过调节温度较低的空气流量带走电堆中热量, 可以达到电堆工作温度的有效控制。但在电堆负载动态跟踪过程中, 由于电堆尾气热量波动变化, 导致被预热空气的电堆入口温度也随之波动。当空气电堆入口温度与电堆温度相差过大时, 相应的热应力很大, 会导致电池片变形和损坏电堆密封性。为此, 美国国家燃料电池研究中心 F. Muller 等人通过在空气供气系统中引入一路流量可调的冷空气管道, 用于调节电堆空气入口温度, 进而控制电堆进出口温差, 并集成了相应的外围供气系统验证了其可行性<sup>[14, 15]</sup>。但是, 通过调节旁路冷空气流量, 这仅仅只能满足预热后进入电堆的空气温度高于需求温度的情况, 如果预热后空气温度低于需求温度, 则无法调节; 另外, 在以上供气系统中, 对于氢气一路没有进行预热处理, 而是直接从重整器输入电堆, 这样将导致电池片阴、阳极两侧温差不可控, 存在电池片破裂的危险。

在控制策略的设计上, 张兄文等人基于模型预测控制, 通过调节电流密度、燃料和空气的摩尔流量控制 SOFC 的输出功率、燃料利用率和电堆温度。该项研究不是从 SOFC 独立系统的角度出发, 而仅以 SOFC 电堆为研究对象, 特别是对电堆温度的控制, 是在电堆入口气体温度恒定这样一个假设前提下完成的<sup>[16]</sup>; K.Sedghisigarchi 将 SOFC 的非线性动态模型线性化, 在此线性模型的基础上设计出了  $H_\infty$  控制器, 使得系统在负载发生变化情况下, 能通过改变输入燃料流速来调节电池的输出电压, 同时将电池利用率控制在一定范围内, 以起到保护电堆的作用<sup>[17]</sup>; P.Auiar 设计了一个主控制器和一个 PID 温度控制器, 其中主控制器用于对 SOFC 电堆电流施加扰动; 反馈 PID 温度控制器通过调节空气流速, 将燃料出口的温度控制在期望值附近<sup>[18]</sup>。但是, 这些控制策略的设计都是将被控对象锁定在 SOFC 系统某个或某几个组成部分上, 没有对系统进行整体性的综合考虑; 同时假定条件过于理

想化，在实际中要满足这些假定条件也不是一件容易的事情；所以这些控制策略都具有局限性，并不能满足实际应用的需要。

在控制策略的实现上，目前国内外通常的做法是将在仿真阶段得到的控制算法通过手工编程的方式转化为嵌入式微处理器或 PLC 等控制执行器能识别的代码，然后再由它们作用于被控对象。一旦发现控制算法不能满足要求，又必须进行重新设计，然后再手工编写代码，如此往复直到控制算法满足要求为止。这种方式有很大的缺陷：不能把注意力集中在控制算法的设计上，对于获得的控制灵感不能马上验证它们的有效性，必须通过编写代码的方式使它们在控制平台上实现，因此浪费了很多不必要的时间在控制算法的代码编程上；而且常常出现的情况是控制算法的失效是由代码编写的错误导致的，从而还要化时间去判断是控制算法选择不当还是代码实现过程的错误，从而又浪费了大量时间。所以这种控制策略的实现方式限制了人的创造力，同时效率低下，开发周期时间长。

为了设计出合理的 SOFC 外围热管理系统，本文在对国内外研究成果和相关文献进行了充分研究的基础上，对上述 SOFC 系统中存在的工艺结构不合理、控制策略存在局限性、控制策略的实现效率低下等缺陷进行了逐一改进。

工艺结构是 SOFC 外围热管理系统中基础，只有设计出合理的工艺结构，然后在以此为基础进行控制策略的研究才会有意义；否则一旦工艺结构设计失当，控制策略再好也不可能取得预期目标。为了解决工艺结构中存在的 SOFC 电堆入口气体温度不可控从而可能导致 SOFC 会因为温差过大而损坏的缺陷，在深入分析了国内外 SOFC 热管理系统工艺结构利弊的基础上，课题组对系统的工艺结构进行了全新改进，改进后的 SOFC 热管理系统分为内部热管理系统和外部热管理系统两部分。内部热管理系统是仅以电堆的工作温度作为被控对象，通过过量的参与 SOFC 电化学反应的空气流量的调节来带走 SOFC 电化学反应放出的热量，从而有效的控制其温度在最佳工作点上。外部热管理系统的设计和控制是内部热管理系统的坚实基础。外部热管理系统是指为了保证内部热管理系统的可控性和有效性，将燃烧器、热交换器、热平衡器等热容对象集成起来构成的换热系统，用于有效管理电堆入口气体温度。在外部热管理系统中，分别进入 SOFC 电堆阴阳两极的两路反应气体由于通

过了热交换器和热平衡器，所以电堆阴阳两极的温差不会太大；同时可以通过燃烧室燃料的调节（一定配比的甲烷和空气），控制反应气体进入电堆的入口温度保持在一个稳定状态（720℃）；这样就解决了上述工艺结构中因为电堆入口气体温度不可控，电堆阴阳两极温差过大或电堆入口温度和电堆本身温度温差过大从而可能导致 SOFC 电堆损坏的缺陷。

控制策略是 SOFC 外围热管理系统的方式和手段，在设计出合理的工艺结构之后，接下来的关键就是在这个工艺结构的基础上建立比较准确的系统模型，然后基于这个模型进行控制策略的研究，从而保障 SOFC 系统安全、高效的运行。为了克服 SOFC 系统中控制策略存在局限性的问题，必须把 SOFC 系统作为一个整体被控对象进行控制策略的研究，在本文工艺结构的基础上就是要同时控制反应气进入电堆的入口温度和电堆的本身的温度。基于这种分析，本文在充分认识了系统工艺结构的基础上，设计出了合理的控制策略实现了对电堆入口气体温度和电堆温度的有效控制。

控制策略的实现是 SOFC 外围热管理系统最终的落脚点，在控制策略设计完成以后，如何把控制策略转移到相关硬件平台上进行实时控制以验证控制策略的有效性将是 SOFC 外围热管理系统走向成熟的关键。为了克服常规控制策略实现方式存在限制人的创造力、效率低下的缺陷，本文采用了基于 MATLAB/RTW 自动代码生成技术，将控制策略直接转化为 XPC 平台或 MPC555 平台上的代码以实现实时控制。这种控制策略的实现方式省去了手工代码编写的环节，从而能够将人的灵感充分调动起来集中在控制策略的设计上，缩短了开发时间，是一种比较前卫和高效的开发方式。

### 1.3 论文的主要工作和章节安排

本文的主要工作是对 SOFC 外围热管理系统进行研究，它包括工艺结构的设计、控制策略的设计、控制策略的实现几个重要部分；其中工艺结构是基础，控制结构是手段，控制策略的实现是落脚点。在充分借鉴国内外 SOFC 热管理系统工艺结构

的基础上，课题组对其工艺结构进行了改进，并按照工艺结构成功集成出了 2.5kw 的 SOFC 外围系统。所以本文将以此工艺结构为基础，重点介绍 SOFC 热管理系统中另外两个重要的组成部分：控制策略的设计和实现。

本文的章节安排为：第 1 章介绍课题研究的意义和国内外发展现状，以及课题组在 SOFC 热管理系统工艺结构方面所取得的进展，还有个人所做的工作；第 2 章从工艺结构进行分析，并结合试验数据运用 T-S 模糊辨识方式辨识出系统各部的模型；第 3 章在上一章建立的模型的基础上，采用 StateFlow 设计逻辑控制策略以满足 SOFC 系统逻辑控制阶段的要求，采用串级 PI 控制策略来满足连续控制阶段的要求，并将这两种控制策略合理统一起来；第 4 章建立 XPC 半实物平台，通过 MATLAB/RTW 将 Simulink 下的仿真控制策略转化为 XPC 平台下的实时控制代码，以验证上一章控制策略的有效性；第 5 章在 MPC555 上进行 UCOS 操作系统的移植，提高其实时性以便取代上一章体积庞大的 XPC 控制平台，提高整个 SOFC 系统的集成度，向产品级 SOFC 独立发电系统看齐；第 6 章对本文所做的工作进行总结和展望。

## 2 热管理系统模型的搭建

为了有效的控制 SOFC 电堆的工作温度，必须设计出合理的 SOFC 热管理系统的工艺结构，在此基础上才有可能对电堆的工作温度进行良好的控制。本文在设计并集成了 1kw 的 SOFC 热管理系统实物后，在现有的阶段，是通过手动调节方式实现对 SOFC 电堆工作温度的控制。为了从手动控制方式过渡到自动控制方式，必须建立 SOFC 热管理系统被控对象的数学模型，以便为控制策略的研究打下基础。为此，本文首先介绍 SOFC 热管理系统的工艺结构，熟悉整个系统运作原理，然后按照这个工艺结构采用 T-S 模型建模方式建立 SOFC 热管理系统被控对象的数学模型，为 SOFC 热管理系统控制策略的研究提供一个良好的基础。

### 2.1 SOFC 系统的工艺结构

为了控制 SOFC 电堆的工作温度必须设计出合理的工艺结构，它是能否进行有效控制的基础。为此，下面将重点介绍 SOFC 热管理系统的工艺结构，SOFC 系统主要由 2 级热交换器、热平衡器和电堆组成，其工艺结构如图 2-1 所示：

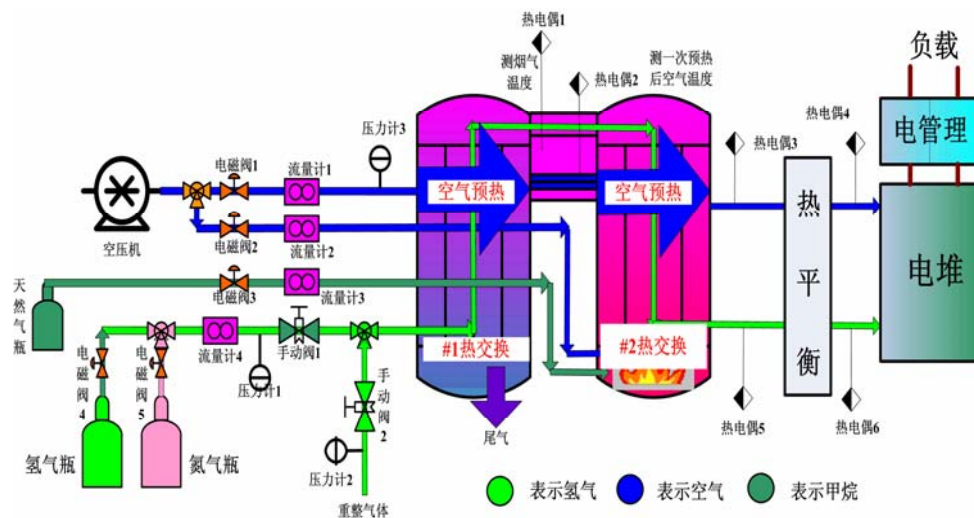


图 2-1 系统工艺结构图

此工艺结构的目的在于：1) 把电堆的工作温度控制在最佳工作点上，即  $750^{\circ}\text{C}$ ；2) 反应气体进入电堆的入口温度控制在  $720^{\circ}\text{C}$  以保证两者之间的温差不会太大以免 SOFC 电堆损坏，同时利用这个温差，通过通入过量参与反应的空气来带走电化学反应放出的热量，保证电堆的工作温度在  $750^{\circ}\text{C}$ 。

为了提高能量的利用率在这里采用了两极热交换的结构：第一级热交换是个粗调过程，由燃烧室燃烧产生的高温废气为参与电化学反应的氢气和空气供热，反应气体的温度经过此次热交换后可以达到  $150^{\circ}\text{C}\sim 300^{\circ}\text{C}$ ；第二级热交换是个细调过程通过燃烧室燃烧的火焰直接对反应气体供热，以便让两路反应气体通过热平衡器之后温度都稳定在  $720^{\circ}\text{C}$ 。热平衡器的作用主要是将两路温度不同的反应气体进行充分换热后达到两者温度相同的目的是，它的引入是为防止两路进入电堆反应气体温差过大而引起电堆阴阳两极温度不均而出现电堆损坏。在这里空气分为两路：一路和甲烷一起进入燃烧室进行燃烧，在二级热交换中为反应气体供热；另一路和氢气一起进入电堆参加电化学反应，释放出电能供给负载。氮气在这里的作用是处于安全性的考虑在 SOFC 系统启动和停机过程中进行吹扫，将杂质气体排出。

上面的工艺结构设计还只是停留在理论设计、分析阶段，为了验证其可行性，按照图 2-1 所示的工艺结构，设计并集成出了 1kw 的 SOFC 热管理系统实物，实物图如图 2-2 所示：

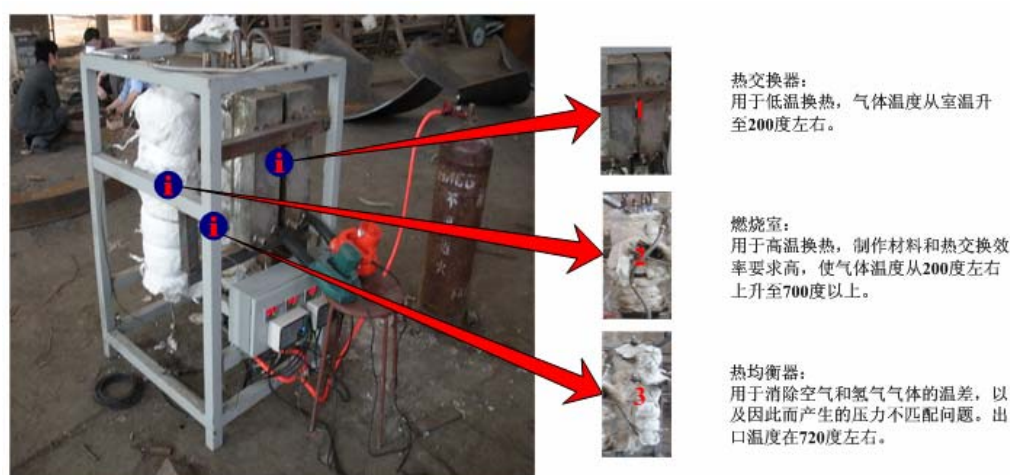


图 2-2 集成的 SOFC 热管理系统实物图

为了验证工艺结构的可行性，在对被控对象一无所知时，通过 FameView 组态软

件手动调节甲烷、空气、氮气、氢气的流量，并将这些流量数据通过串口通讯的方式发到 PLC 上来控制电堆入口空气温度和电堆温度，实验结果证明这种工艺结构的设计是完全合理、可行的，可以有效的控制电堆入口空气温度和电堆温度。FameView 组态软件调节监控图如图 2-3 所示：



图 2-3 FameView 组态软件调节监控图

上面是通过手动调节的方式来控制电堆入口空气温度和电堆温度，为了满足实际应用的需要，必须把手动调节方式替换为自动调节方式，为了达到这个目的，必须先把上述的定性的工艺结构和流程转化为定量的数学模型，然后在此基础上进行控制策略的研究，实现对电堆本身温度和电堆入口温度有效、合理的自动控制。

## 2.2 建模方法简介及选择

为了建立 SOFC 系统的数学模型，必须采用合适的建模方法，目前通常有两种建模方法：机理建模法和测试建模法。

机理建模法就是根据事物内在运行机制或变化规律，写出各种有关的平衡方程如：物质平衡方程，能量平衡方程，动量平衡方程以反映这种机制或规律，从而获得所需的数学模型<sup>[19]</sup>。因此机理建模运用的前提条件应该对需要建立模型的对象有

充分的认识，可以采用相关的数学公式来进行描述对象的行为特性。测试建模法是将待建模的对象当做一个黑箱来处理，仅关心它的输入输出特性，完全用外部特性来表征对象的行为，不需要深入了解对象的内部机理。

课题组的曹红亮博士曾经做过 SOFC 热管理系统的机理建模，但是为了得到相对简单的数学表达式，建立的模型做了很多理想化的假设和简化，得到的机理模型与按照图 2-2 和图 2-3 进行实验所获得的实验数据不能很好的匹配，为此必须对 SOFC 热管理系统的模型进行重新设计。由于本文中 SOFC 系统的工艺结构复杂，要准确把握其内部各部分的运行特性是一件十分困难且费力的事情，但是却可以通过手动调节实验，利用 FameView 组态软件就可以很容易记录下甲烷、空气、氮气、氢气的流量以及电堆入口空气温度和电堆温度等相关的输入、输出信息，因此可以选用测试建模法来建立 SOFC 系统各部分的模型，然后再按照工艺结构将它们统一成一个整体，从而为 SOFC 热管理系统控制策略的设计打下基础。在众多测试建模方法中，采用 T-S 模糊建模方法建立的 T-S 模糊模型由于具有非常明显的特色，因此得到了广泛应用。

### 2.3 SOFC 系统的 T-S 模糊建模

Takagi 和 Sugeno 在 1985 年提出了一种 T-S 模糊模型，它是一种本质非线性的模型，它以局部线性化为基础，通过模糊推理实现全局的非线性<sup>[20, 21]</sup>。在采用模糊模型对系统进行建模时，可以不知道系统内部的具体工作机制，它通过模糊推理方式构造出输入与输出之间的一种映射关系，十分适用于大时滞、非线性、时变、强耦合的复杂系统。由于 SOFC 系统就是这么一个大时滞、非线性、强耦合的复杂系统，所以采用 T-S 模糊模型来描述系统各部分的输入输出关系是再适合不过了。

T-S 模糊模型是基于输入模糊划分的思想，是分段线性化的拓展。一个典型的 MISO 动态系统如果用 T-S 模糊模型表示，可以表示为：

Ri : if  $x_1$  is  $A_{1,i}$ ,  $x_2$  is  $A_{2,i}$ , ...,  $x_m$  is  $A_{m,i}$

$$\text{Then : } y_i = p_0^i + p_1^i x_1 + p_2^i x_2 + \dots + p_m^i x_m = p_0^i + \sum_{j=1}^m p_j^i x_j, i = 1, 2, \dots, n \quad \text{式(2-1)}$$

其中,  $R_i$  表示第  $i$  条模糊规则;  $x_i$  是当前输入样本中的第  $i$  个输入变量, 是构成规则前提的变量;  $A_{j,i}$  是第  $i$  条规则中第  $j$  个输入变量的模糊子集, 其隶属函数中的参数称为前提参数;  $p_j^i$  是第  $i$  条模糊规则结论中的第  $j$  个输入的参数;  $y_i$  是第  $i$  条模糊规则的输出;  $m$  是输入变量的个数;  $n$  为模糊规则的总数。由上面的  $n$  条模糊规则可以得到  $k$  时刻模糊模型的输出为:

$$y(k) = \sum_{i=1}^n \lambda_i * y_i(k) = \sum_{i=1}^n \lambda_i * (p_0^i + p_1^i x_1(k) + p_2^i x_2(k) + \dots + p_m^i x_m(k)) = \phi^T(k)\theta,$$

$$\phi^T(k) = [\lambda_1, \lambda_2, \dots, \lambda_n, \lambda_1 x_1, \lambda_2 x_1, \dots, \lambda_n x_1, \dots, \lambda_1 x_m, \lambda_2 x_m, \dots, \lambda_n x_m]$$

$$\theta = [p_0^1, \dots, p_0^n, p_1^1, \dots, p_1^n, p_m^1, \dots, p_m^n]$$

$$\lambda_i = \frac{\mu_i}{\sum_{i=1}^n \mu_i}, \quad \mu_i = \prod_{j=1}^m A_j^i(x_j) \quad \text{式(2-2)}$$

其中,  $\lambda_i$  是第  $i$  条模糊规则在总规则中所占的权重,  $A_j^i(x_j)$  是第  $j$  个输入变量在第  $i$  条模糊规则中所对应模糊子集的隶属度,  $\theta$  是后件参数,  $\prod$  是模糊算子, 通常取极小或乘积运算。

为了建立 T-S 模糊模型, 必须对它的参数进行辨识。T-S 模糊模型辨识的任务是找到输入和输出之间通过模糊规则对应起来的一种映射关系, 其内容包括两部分: 1 前件结构和参数辨识, 2 后件参数辨识。

#### 1) 前件结构和参数辨识:

通过模糊 C 均值聚类算法, 把输入的样本  $\{x_1, x_2, x_3, \dots, x_n\}$  分成  $C$  个模糊类, 其中  $x_i$  在这里是某一时刻的输入向量, 第  $k$  个数据向量属于第  $i$  个模糊聚类的隶属度函数用  $\mu_{ik}$  表示。设定目标函数为:

$$J = \sum_{k=1}^n \sum_{i=1}^c (\mu_{ik})^m \|x_k - c_i\|^2 \quad \text{式(2-3)}$$

其中,  $m$  是模糊化参数,  $1 \leq m \leq \infty$ , 将目标函数极小化可得到下面的聚类算法。

(1) 给定  $m$  和  $c$ , 对所有  $n$  个样本点随机分配它们对第  $i$  个模糊聚类的隶属度  $\mu_{ik}$

(2) 计算模糊聚类中心  $c_i$  :

$$c_i = \frac{\sum_{k=1}^n (\mu_{ik})^m x_k}{\sum_{k=1}^n (\mu_{ik})^m}, i = 1, 2, \dots, c \quad \text{式(2-4)}$$

(3) 更新  $\mu_{ik}$  :

$$\mu_{ik} = \left[ \sum_{j=1}^c \left( \frac{\|x_k - c_i\|}{\|x_k - c_j\|} \right)^{\frac{2}{m-1}} \right]^{-1}, i = 1, 2, \dots, c; k = 1, 2, \dots, n \quad \text{式(2-5)}$$

(4) 如果  $\|\mu^{new} - \mu\| < \varepsilon$ ,  $\varepsilon$  为一非常小的数, 则算法停止, 否则进入(2)。

经过模糊聚类算法后, 确定了聚类中心和每个样本点关于聚类中心的隶属度, 其中聚类中心与模糊规则一一对应, 每个聚类中心就对应了相关的模糊规则, 每个样本点的隶属度也就是每条模糊规则的对它适用程度, 也就是上文讲到的每条模糊规则在总规则中所占的权重  $\lambda_i$ , 接下来所要做的工作就是进行后件参数的辨识。

2) 后件参数辨识:

由式(2-2)可得  $y(k) = \phi^T(k)\theta$ , 若将  $k$  个时刻的输入和输出关系用矩阵的形式表示, 可得:  $Y = \Phi \cdot \theta$ 。其中:

$$Y = [y(1), y(2), \dots, y(k)]^T$$

$$\Phi = \begin{bmatrix} \phi^T(1) \\ \phi^T(2) \\ \vdots \\ \phi^T(k) \end{bmatrix} \quad \text{式(2-6)}$$

采用最小二乘法可得:  $\theta = (\Phi^T \Phi)^{-1} \Phi^T Y$ 。到此, 就得到了一个 T-S 模糊模型。

为了更好的反映实际的工艺结构让 SOFC 系统模型有更高的可信度, 以便更有效的设计控制策略实现系统温度控制的需要, 在这里需要对热交换器 1、热交换器 2、热平衡器、电堆进行分别建立相关的温度模型。由于热交换器 1 和热交换器 2 实际上都是由燃料的燃烧来供热并且二者结合紧密, 因此可以将它们作为一个整体称为

燃烧器进行统一建模，这样就可以在不影响模型可信度的前提下简化模型结构。

在充分分析和处理系统试验数据的基础上，应用 T-S 模糊模型来分别描述 SOFC 系统中燃烧器、热平衡器、电堆这三个重要组成部分的温度特性对试验数据进行拟合，拟合效果如图 2-4 至图 2-9 所示：

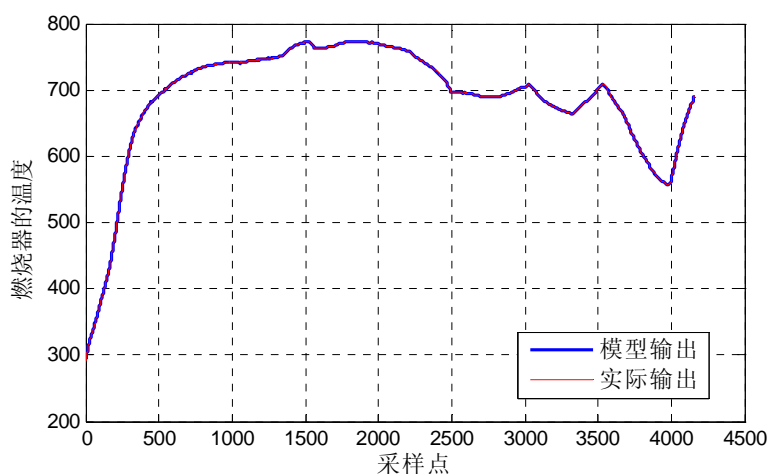


图 2-4 燃烧器的温度拟合曲线

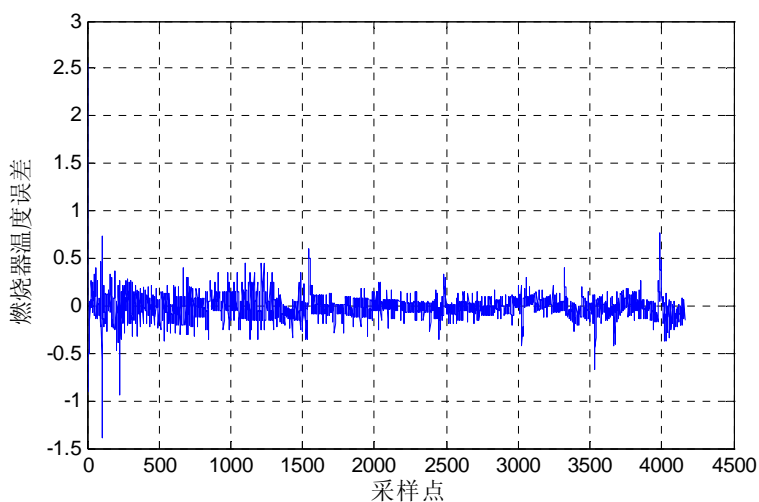


图 2-5 燃烧器温度的拟合误差曲线

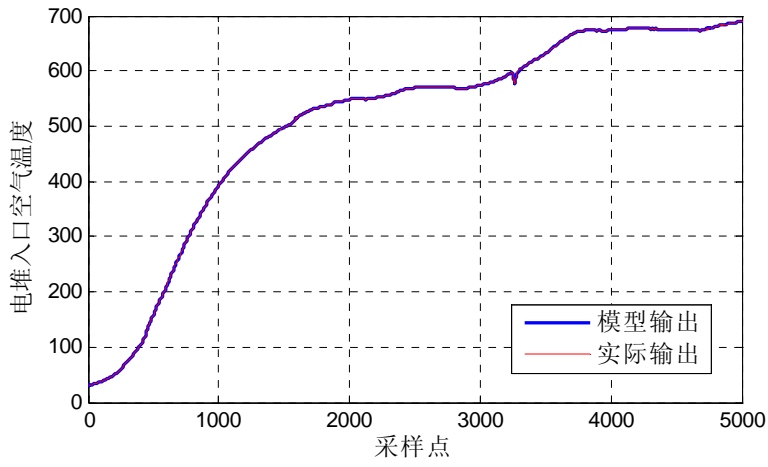


图 2-6 电堆入口空气温度拟合曲线

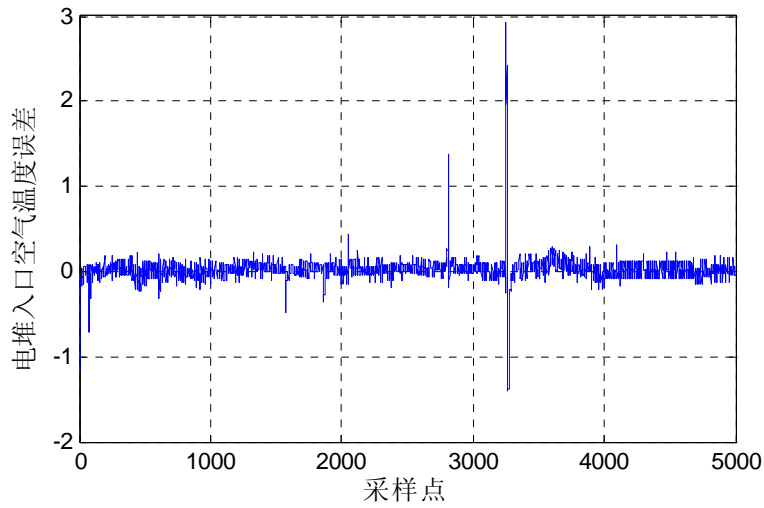


图 2-7 电堆入口空气温度拟合误差曲线

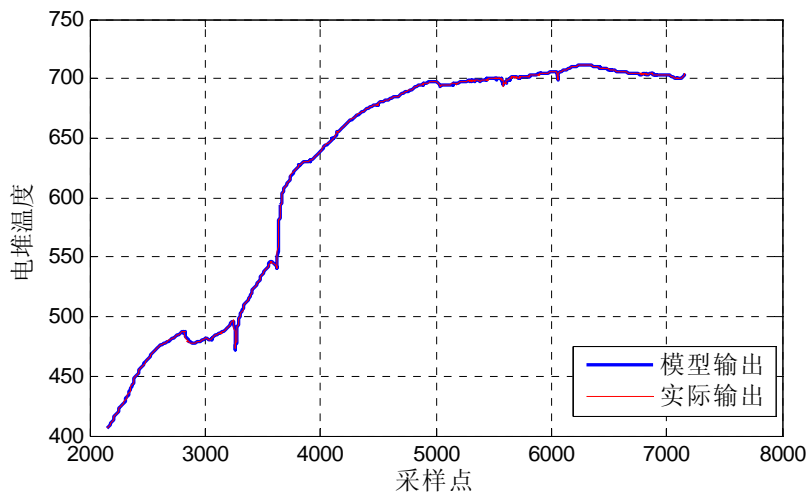


图 2-8 电堆温度拟合曲线

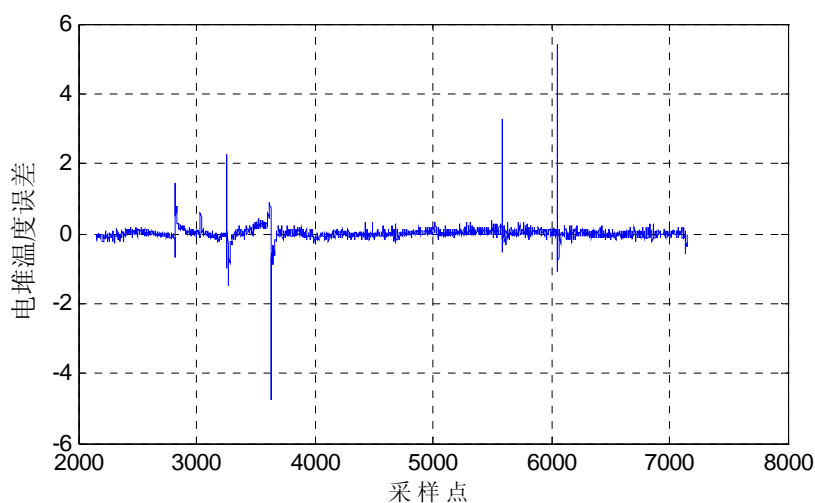


图 2-9 电堆温度拟合误差曲线

从上面图形可以看出 T-S 模糊模型取得了不错的建模效果，可以比较准确的描述系统各个组成部分的温度特性。

## 2.4 本章小结

本章首先详细的介绍了整个 SOFC 系统的工艺结构和流程；然后在此基础上确定出系统模型结构由热交换器 1、热交换器 2、热平衡器、电堆这 4 部分组成；最后通过分析将系统模型结构进行了合理简化，并采用 T-S 模糊建模方法对简化后的系统模型各部分分别进行建模，通过比较，发现建立的模型能够比较准确的描述各部分的温度特性，从而为控制策略的设计奠定了良好的基础。

### 3 热管理系统控制策略的研究

在上一章中已经对 SOFC 系统工艺结构的各个组成部分建立了比较准确的温度模型，在本章中将以此为基础对热管理系统的第二个重要组成部分即控制策略进行研究。按照操作规程，在 SOFC 系统发电过程中有两个阶段：预热阶段和发电阶段。在预热阶段，需要根据电堆入口空气温度逐步加大氮气和空气对 SOFC 系统进行预热，以保证系统的安全性同时让系统达到一个较高的温度状态以保证电化学反应法发生时具有较高的能量转化效率；当电堆入口空气温度超过某一温度后，需要关闭氮气并通入氢气，系统进入发电状态，在这个阶段需要控制电堆入口空气温度在  $720^{\circ}\text{C}$  同时电堆温度在  $750^{\circ}\text{C}$ ，以保证系统安全、高效的运行。对应操作规程，控制策略的设计应该对应分为两个部分：在预热阶段应该设计满足操作规程的逻辑控制器；在发电阶段应该设计满足要求的连续动态控制器，同时控制电堆入口空气温度和电堆温度。

#### 3.1 逻辑控制过程控制器的设计

按照操作规程，预热阶段的详细过程为：当电堆入口空气温度在  $0\sim 150^{\circ}\text{C}$  时，通入  $10\text{L}/\text{min}$  的甲烷和一定配比的空气到燃烧器进行燃烧；当电堆入口空气温度在  $150\sim 300^{\circ}\text{C}$  时，再通入  $10\text{L}/\text{min}$  的空气和  $3\text{L}/\text{min}$  氮气到 SOFC 电堆中，对电堆进行预热；当电堆入口空气温度在  $300\sim 450^{\circ}\text{C}$  时，预热空气的流量变为  $20\text{L}/\text{min}$ ，氮气的流量变为  $5\text{L}/\text{min}$ ；当电堆入口空气温度在  $450\sim 650^{\circ}\text{C}$  时，预热空气的流量变为  $30\text{L}/\text{min}$ ，通入氮气的流量变为  $15\text{L}/\text{min}$ ；在电堆入口空气温度超过  $650^{\circ}\text{C}$  时，预热阶段结束，系统进入发电阶段。

要实现上述预热阶段的控制过程，可以采用 PLC 来实现。目前对于复杂的逻辑控制过程，常常利用 PLC 来进行逻辑控制。PLC 简称可编程逻辑控制器，它采用可编程的存储器，用来在内部存储逻辑运算、顺序控制、定时、计数和算术运算等操

作的指令；通过数字式、模拟式的输入和输出控制各种类型的机械或生产过程。由于 PLC 具有 1)可靠性高，抗干扰能力强，2)通用性能强，控制程序可变，使用方便 3)编程简单，容易掌握 4)体积小、重量轻、功耗低、维护方便等特性，在工业控制领域得到了广泛的应用<sup>[22]</sup>。

对于逻辑控制策略的设计，PLC 广泛采用梯形图的实现方式。这种方式具有直观易懂，很容易被现场操作人员掌握的优点；但是这种方式也有其固有的缺陷：缺乏层次性，所有的逻辑关系都处于同一层次上，一旦发现某个地方逻辑不够合理需要修改，那么其他地方也必须相应进行修改，所以梯形图的设计和修改繁琐，需要瞻前顾后。为了方便、快捷的进行逻辑控制策略的设计，同时保留直观易懂的特性，本文采用 StateFlow 来对 SOFC 热管理系统的预热阶段进行逻辑控制。

StateFlow 是集成于 Simulink 中的图形化设计与开发工具，主要用于对复杂逻辑系统进行建模和仿真，与复杂逻辑系统对应的是动态变换系统。动态变换系统通常利用数学表达式、方程等组成的输入和输出关系进行描述；而复杂逻辑系统通常利用一些自然语言或者逻辑表达式进行描述<sup>[23]</sup>。采用 StateFlow 来进行建模和进行逻辑控制器的设计，可以利用 StateFlow 可视化的模型和直观的仿真能力，清晰、简洁地反映出复杂动态逻辑关系。StateFlow 可以嵌入到一般 Simulink 模型中，从而构成事件响应系统和动态变换系统组合而成的复杂系统。

在复杂逻辑系统中，系统会从一种操作模式转移到其它的操作模式上，因此为了表示复杂逻辑系统的操作模式，StateFlow 提出了状态的概念。状态是 StateFlow 的核心，一种状态可以表示一种操作模式，在任意时刻 StateFlow 中的状态要么处于活动状态，要么处于非活动状态，不肯能出现第三种状态，所以可以把状态看做高级编程语言中的布尔数据类型。状态在系统中被看成记忆元件，它本身可以保持活动的模式，一旦被激活，状态就保持活动的模式，直到系统改变其模式，状态才变为非活动的；同理，如果状态是非活动的，则状态就会一直保持非活动的状态，直到系统改变其工作模式为止。在 StateFlow 中存在并行状态和互斥状态两种状态。并行状态是只同一时刻可以同时处于激活的状态；互斥状态是指在同一时刻不可能同时激活的状态。

和 Simulink 一样，在 StateFlow 中状态是有层次性的，在同一层次上所有的状态要么是互斥的要么是并行的，不可能出现既互斥又并行的状态。由于层次性的存在，在修改某个状态的时候只需要修改它本身的状态和子状态即可，与其处于同一层次的状态和父状态则可保持不变，这样使 StateFlow 使用起来更加灵活、方便，所以其具有比 PLC 中梯形图设计更明显的优势。

为了实现预热阶段的控制过程，在这里采用 StateFlow 来进行控制器的设计。结合前一章得到的 SOFC 热管理系统模型，StateFlow 下控制器的设计过程如下：

#### 1) 定义 StateFlow 与 Simulink 的接口

由于操作规程是按照电堆入口空气温度的值来进行控制动作的，所以电堆入口空气温度是 StateFlow 的一个输入，另一个输入实际过程中的启动信号；同时由于操作规程控制的气体流量包括甲烷、空气、氮气，所以 StateFlow 的输出是甲烷、空气、氮气等气体的流量。

#### 2) 定义所需的状态及其结构

由于 SOFC 热管理系统包括启动、停机等 2 个互斥结构的状态，它们应该处于 StateFlow 中的第一层。同时在启动状态中应该包括电堆入口空气温度在 0~150℃，电堆入口空气温度在 150~300℃，电堆入口空气温度在 300~450℃，电堆入口空气温度在 450~650℃，电堆入口空气温度在 650℃ 以上等 5 个互斥结构的状态，它们是启动状态的子状态，处于 StateFlow 中的第二层。

#### 3) 定义状态动作及变量

当停机发生时，系统的所有输出都归零，所以 StateFlow 停机状态的动作为所有气体流量值都为 0；在启动状态下，其各个子状态根据操作规程来定义自己的状态动作，如在电堆入口空气温度在 0~150℃ 的子状态，状态动作为输出 10L/min 的甲烷，电堆入口空气温度在 150~300℃ 的子状态，状态动作为输出 10L/min 的空气和 3L/min 的氮气，以此类推。

#### 4) 定义状态之间的转移和默认转移

因为任意两个互斥状态不可能在同一时间被激活，所以需要添加转移来控制从一个互斥状态到另一个互斥状态之间的切换；而并行状态是同时激活的，所以它们

之间不需要用转移来进行切换。转移是 StateFlow 中最常见的图形元素之一，当转移发生时，原状态变为非活动状态，目标状态变为活动状态。在本文的 StateFlow 逻辑控制结构中，启动和停机状态之间的切换是通过外部启动/停止信号进行控制的；而启动状态内部的 5 个互斥操作状态之间的切换是通过电堆入口空气温度来进行控制的。

默认转移是一类特殊的转移，可以把它看做是状态转移的特例。默认转移确定多个互斥状态的父状态处于激活态时，在其所有子状态中第一个被激活的状态，在 StateFlow 中必须为每个层次的互斥状态添加默认转移。在本文的 StateFlow 逻辑控制结构中，停机状态应该是第一层次的默认转移状态；启动状态的子状态电堆入口空气温度在 0~150℃的状态应该为第二层次的默认转移状态。到此，SOFC 热管理系统预热阶段的逻辑控制过程的设计就完成了，完成后的 StateFlow 控制器的内、外部结构图如图 3-1 和图 3-2 所示：

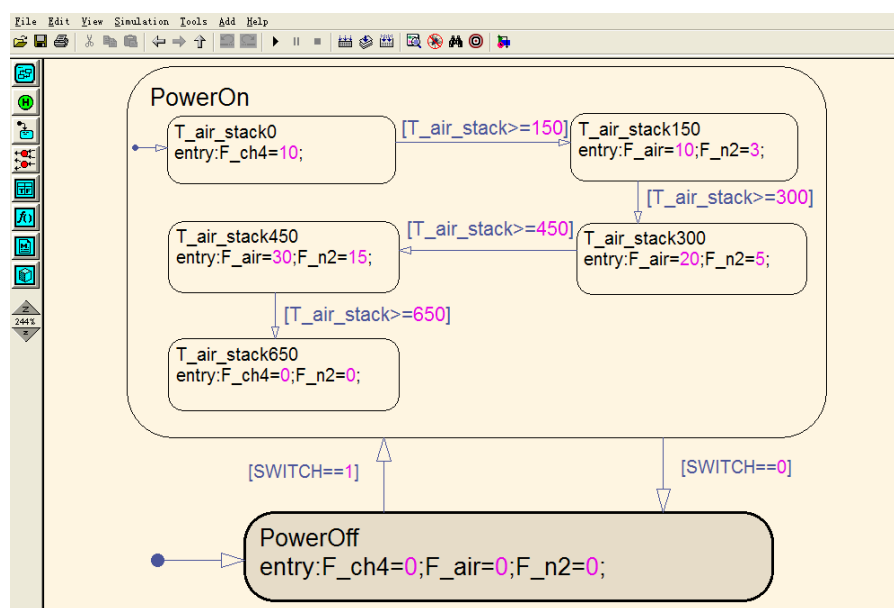


图 3-1 StateFlow 控制器的内部结构图

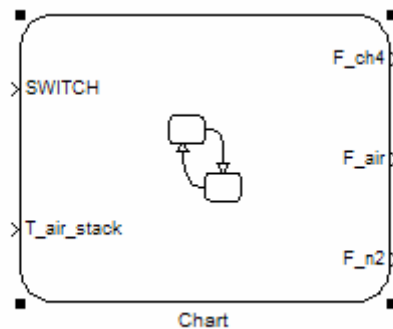


图 3-2 StateFlow 控制器的外部结构图

### 3.2 连续动态过程控制器的设计

当预热阶段结束后,SOFC 系统进入发电阶段。在发电阶段为了让系统处于安全、高效的运行状态,必须控制电堆入口空气温度在  $720^{\circ}\text{C}$ ,电堆温度在  $750^{\circ}\text{C}$ ,为此必须设计有效、合理的连续动态过程的控制策略。

PID (Proportional Integral Differential) 控制是比例积分微分控制的简称。在 20 世纪 40 年代以前,除了在最简单的情况下可采用开关控制外,它是唯一的控制方式。此后,随着科学技术的发展,涌现出许多先进的控制方法,然而直到现在,PID 控制由于它自身的优点仍然是应用最广泛的基本控制方式,占整个工业过程控制算法的  $85\% \sim 90\%$ <sup>[24]</sup>。

PID 控制拥有诸多优点: 1) 适应性强,可以广泛应用于化工、热工、冶金、炼油以及造纸、建材等各种生产部门,按 PID 控制进行工作的自动调节器早已商品化,但是它们仍然是以基本 PID 控制为基础来进行控制参数的自动调节。2) 具有一定的鲁棒性,其控制品质对被控对象特性的变化不大敏感。3) 对模型依赖少,在不完全了解一个系统和被控对象,或不能通过有效的测量手段来获得系统参数时,控制理论的其它技术就难以采用,这时应用 PID 控制成了最佳选择。在实际应用中由于微分作用可能会引入不必要的干扰,所以 PI 控制方式应用更为广泛,因此本文先对 SOFC 热管理系统发电阶段的连续动态过程进行 PI 控制器的设计。

PI 控制算法分为位置式和增量式两种,在本文中由于采用电磁阀来控制气体流

量，因此位置式对应阀门的开度，增量式对应阀门开度的变化。为了防止控制器故障引起的控制量大幅变化，从而导致电磁阀的开度大幅变化，造成电磁阀的损坏，同时也基于减小电磁阀动作误差的需要，在这里采用数字增量型 PI 控制算法，使控制器在上一采样周期阀门开度的基础上，根据误差控制本次采样周期阀门开度的变化，其算法如公式 3-1 所示：

$$\Delta u(k) = K_p * (e(k) - e(k-1)) + K_i * e(k) \quad \text{式(3-1)}$$

Simulink 下的增量型 PI 控制器的设计如图 3-3 所示：

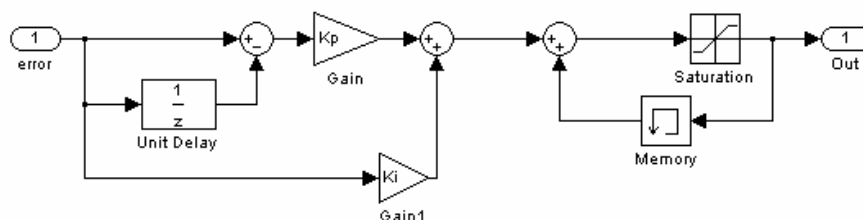


图 3-3 Simulink 下数字增量式 PI 控制器的设计

在第一章的工艺结构改进部分已经介绍过为了在 SOFC 热管理系统中发电阶段同时控制电堆入口空气温度和电堆温度，可以通过分别调节甲烷的流量和参加电化学反应的的空气的流量来实现，本文采用两个增量式 PI 控制器对这两个流量进行调节，并用衰减曲线整定法进行 PI 控制器的参数整定。经过整定后得到调节甲烷流量的 PI 控制器参数为： $K_p=0.6, K_i=0.002$ ；调节空气流量的 PI 控制器参数为： $K_p=2, K_i=0.005$ 。SOFC 系统发电阶段 PI 控制结构图如图 3-4 所示，Simulink 下的实现如图 3-5 所示，控制效果如图 3-6 和图 3-7 所示：

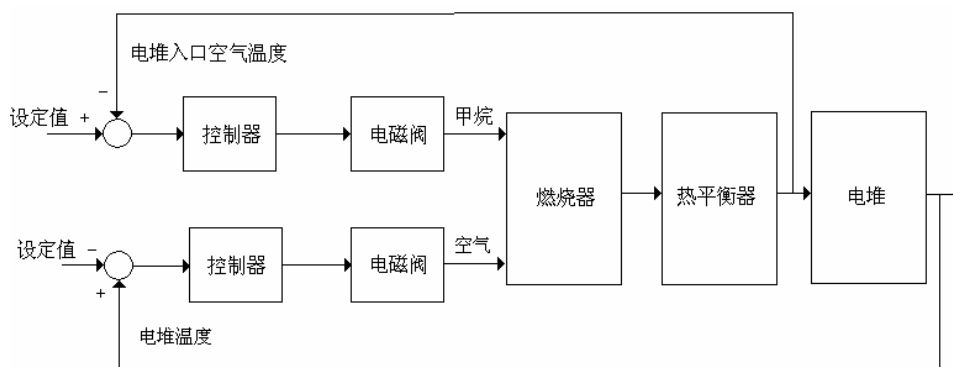


图 3-4 系统 PI 控制结构图

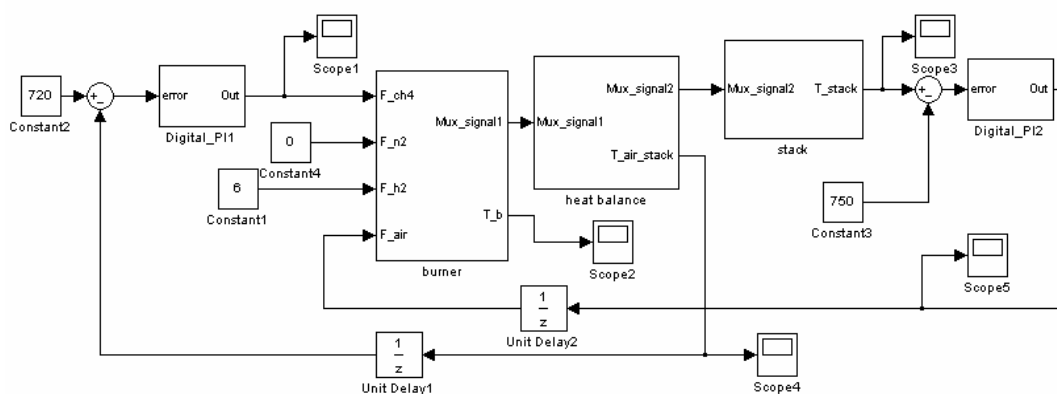


图 3-5 Simulink 下系统 PI 控制实现图

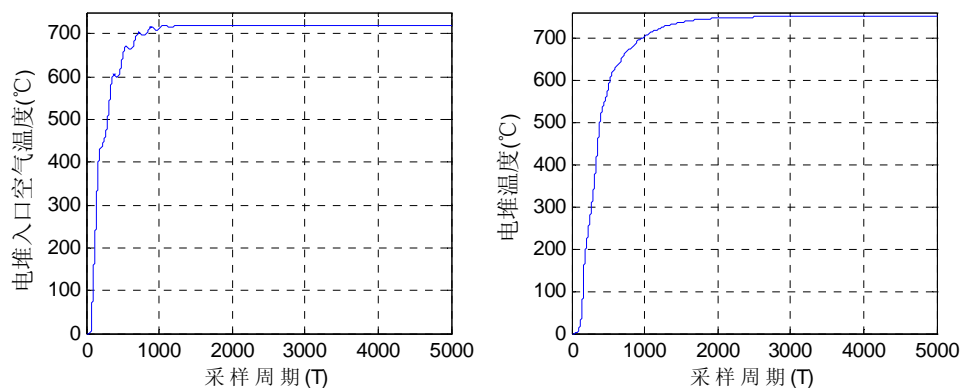


图 3-6 温度变化曲线图

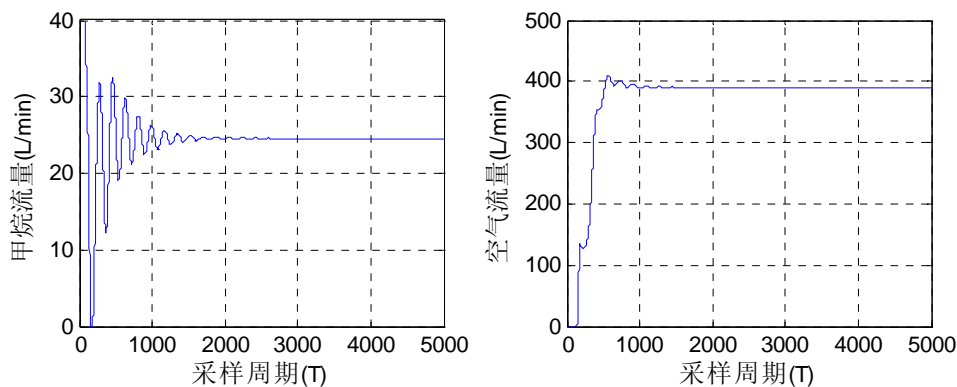


图 3-7 流量变化曲线图

从图 3-6 和图 3-7 可以看到采用 PI 控制取得了不错的控制效果，但是这还不足以说明问题，这是在负载恒定的情况下获得的控制效果，还必须考虑负载发生改变时控制系统的抗干扰能力。经过多次试验发现，电堆的温度随着负载电流的增大而上升，所以必须有效的抑制负载电流发生变化时对 SOFC 热管理系统中电堆入口空气

体温度和电堆温度产生的影响。为了测试系统的抗干扰能力，假设在 Simulink 仿真控制模型中第 4000 个采用周期，由于负载电流的突然增大电堆的工作温度陡升了 50℃，Simulink 下系统抗干扰能力的测试结构图如图 3-8 所示，扰动测试结果如图 3-9 所示：

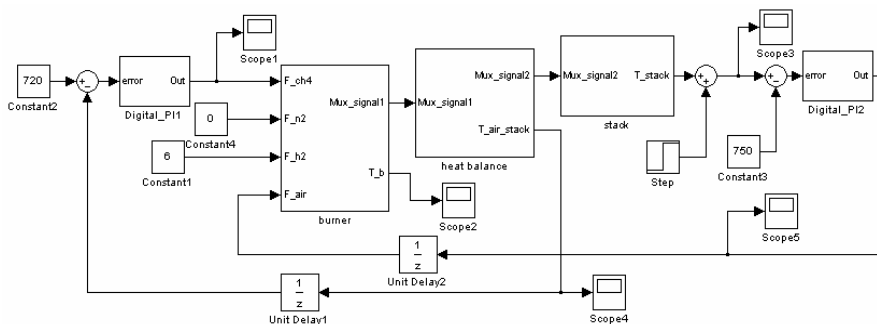


图 3-8 系统抗干扰能力测试结构图

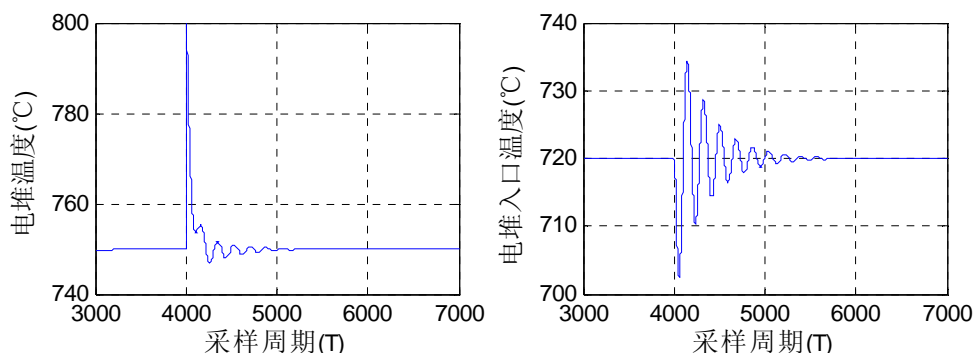


图 3-9 扰动测试结果图

从图 3-9 的扰动测试结果可以看出电堆温度的陡然升高会导致电堆入口空气温度的突降。进过分析这个过程应该这样进行解释：在电堆温度陡然升高时，由于控制作用的存在会通入更多过量的空气来带走电堆的热量以降低电堆的温度；而由于空气流量突然增大，需要预热的气体流量也就更多，所以燃烧器的温度下降，从而导致这些预热气体从燃烧器出来后温度也会相应下降；由于预热的反应气体总体温度下降了，所以从热平衡器出来进入电堆的空气温度下降也就是必然的；由于电堆入口空气温度的下降，在控制作用下会加入更多的甲烷来供热，这样经过一个调整时间后，系统就重新进入了平衡态。

由于在电堆温度陡升时，电堆入口空气温度会突降，有可能因为这个温差过大

引起电堆的损坏，所以在电堆温度突然升高时，必须有效的抑制电堆入口温度的下降，以保证电堆的安全。在图 3-9 中可以看到在电堆温度陡升时，电堆入口空气温度的降幅较大，系统的抗干扰能力不强，并且电堆入口空气温度重新进入平衡态的调节时间过长，在负载电流发生频繁波动的情况下不可能很好的控制电堆温度和电堆入口空气温度，因此必须对上述设计的控制策略进行改进。

通过上述的分析可知，控制系统抗干扰能力不强的主要原因是由于 SOFC 热管理系统是个大时滞系统。当空气的流量突然加大时，这个扰动作用要通过燃烧器、传输管道、热平衡器等多个热容积以后才会导致电堆入口温度的下降，而只有这时电磁阀才开始动作，加大甲烷流量升高燃烧器温度来抑制这个干扰，由于在这段漫长的时间内，系统没有施加任何控制动作，电堆入口空气温度会一直下降，这样也就不难解释图 3-9 中电堆入口空气温度降幅较大的原因了。与此同时电磁阀的调节作用也同样要经过燃烧器、传输管道、热平衡器等多个热容积后才会改变电堆入口空气温度，这时电堆入口温度已经离设定值很远了。也就是说，即使发现了偏差，也得不到及时的调节从而导致系统超调量大，稳定性下降，这也是为什么图 3-9 中发生扰动后电堆入口空气温度调节时间很长的原因。

为了有效的抑制参与反应的空气流量的变化给电堆入口空气温度造成的影响，关键在于尽早的发现干扰的存在，从而进行及时的调节。由于空气流量的加大，最早的反映是燃烧器温度的降低，然后再通过传输通道、热平衡器传递到电堆入口空气温度上，因此可以把燃烧器的温度直接反馈给控制器以缩短延迟时间，让控制器进行及时调节。基于这个分析，可以对电堆入口空气温度的控制在 PI 控制策略的基础上采用串级控制结构来提高系统的抗干扰能力。串级控制下，SOFC 热管理系统的控制结构图如图 3-10 所示：



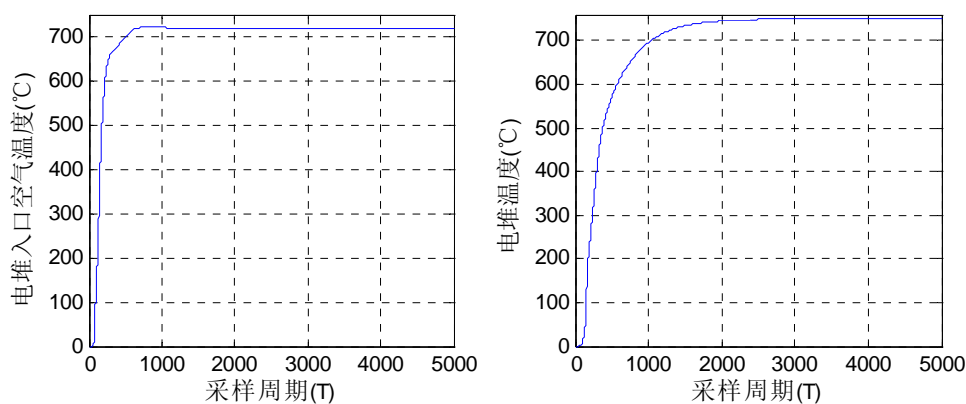


图 3-12 无扰动时温度的变化曲线图

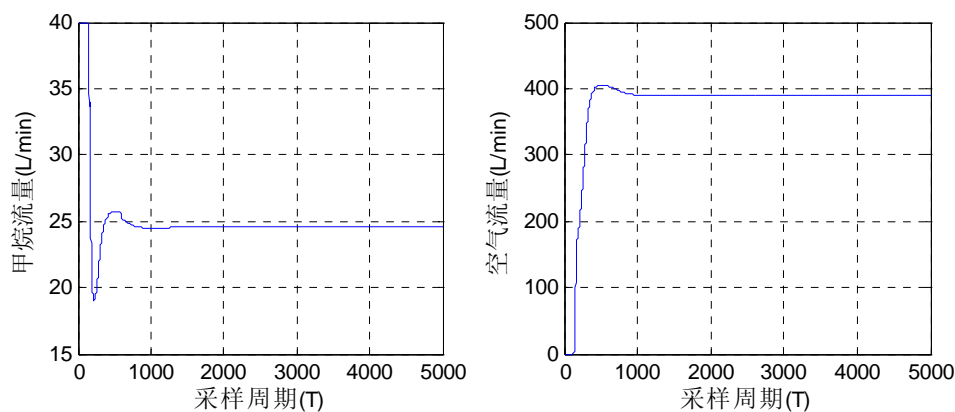


图 3-13 无扰动时流量的变化曲线图

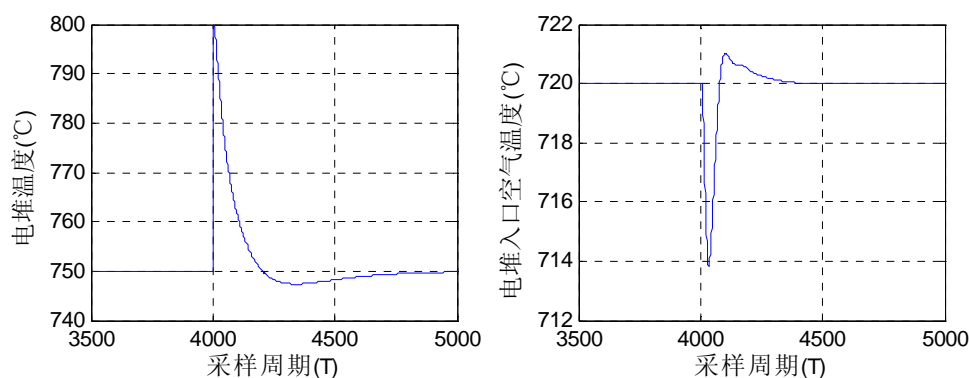


图 3-14 扰动后的温度变化曲线图

从图 3-12 和图 3-13 无扰动时的控制效果，以及图 3-14 的扰动测试结果来看，电堆入口空气温度的串级控制结构显示出比单环控制结构明显的优势，具体比较如表 3-1 所示：

表 3-1 串级 PI 控制与单环 PI 控制的控制效果对比表

	单环 PI 控制	串级 PI 控制
上升时间	约 1800 个采样周期	约 800 个采样周期
超调量	0.7%	0.9%
扰动作用下的温降	18℃	6℃
扰动后的调节时间	约 2000 个采样周期	约 400 个采样周期

从表 3-1 显示串级控制结构能够有效的抑制空气流量发生的变化对电堆入口空气温度的产生的干扰，同时系统的响应性能得到了极大地提升，成功解决了普通 PI 控制中存在的抗干扰能力不强，动态响应能力不足的问题，在负载电流发生频繁波动的情况下也可以实现对电堆温度和电堆入口温度的有效控制，同时由于上升时间的缩短，可以有效的提高能源的利用率。

### 3.3 逻辑控制与连续控制的统一

在 3.1 节和 3.2 节中分别对 SOFC 热管理系统预热阶段和发电阶段分别设计了满足要求的 StateFlow 逻辑控制器和 PI 串级控制器，但是上面的设计过程是独立进行的，为了满足实际应用的要求必须把它们统一起来，并且在预热阶段完成后进行控制器的切换。由于在 StateFlow 逻辑控制器中包含了启动、停止状态，所以可以由逻辑控制器来对 PI 串级控制器进行控制，两种控制器统一后整个 SOFC 热管理系统的结构图如图 3-15 所示，StateFlow 逻辑控制器内部结构图如图 3-16 所示，PI 串级控制器内部结构图如图 3-17 所示：

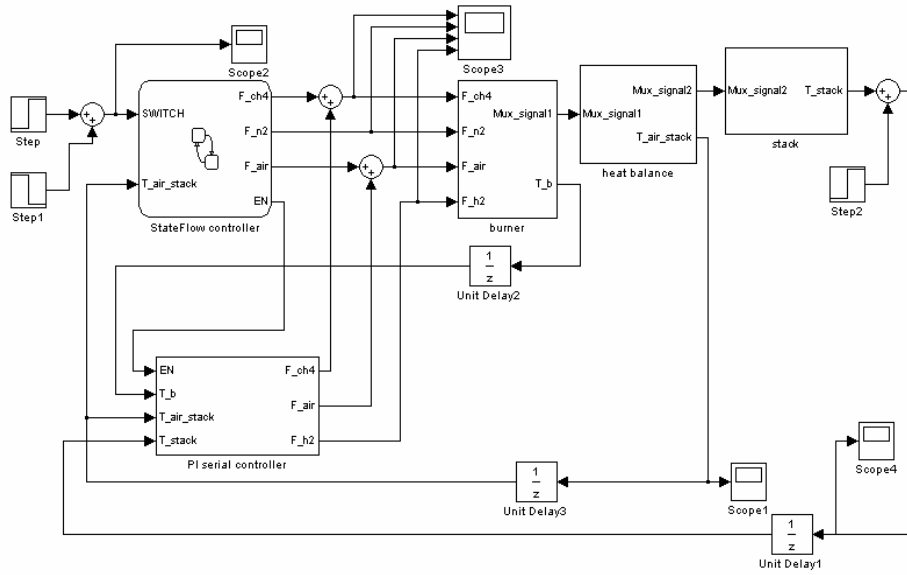


图 3-15 统一后的 SOFC 热管理系统结构图

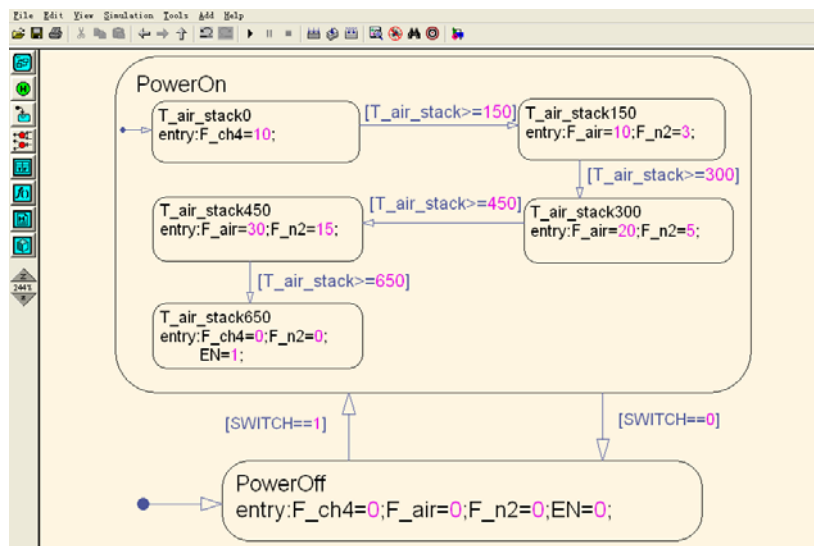


图 3-16 统一后的 StateFlow 逻辑控制器内部结构图

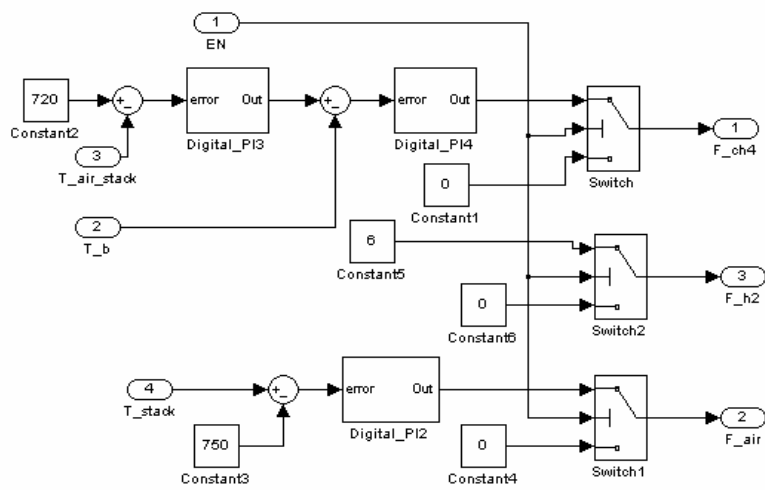


图 3-17 统一后的 PI 串级控制器内部结构图

两种控制策略统一后，仿真时间设为 3000 个采样周期，控制效果如图 3-18 和图

3-19 所示：

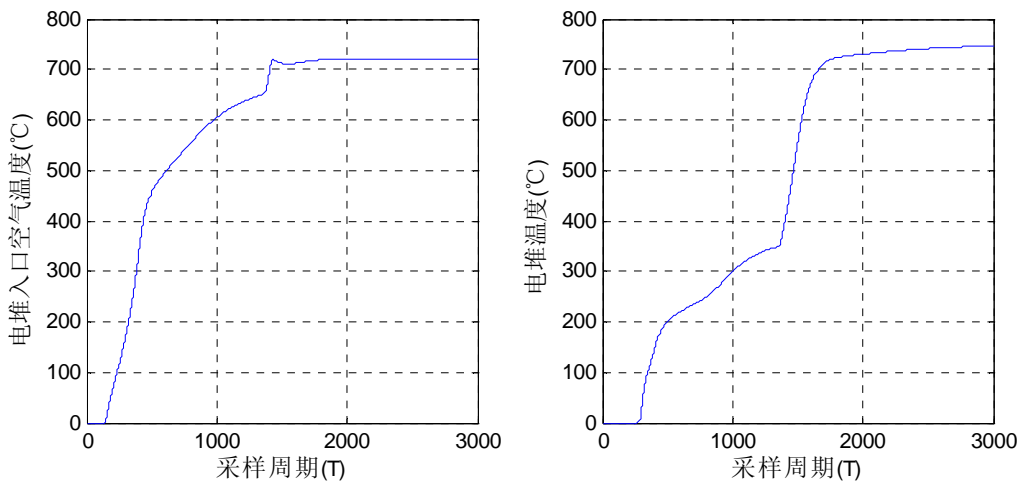


图 3-18 统一后的温度变化曲线图

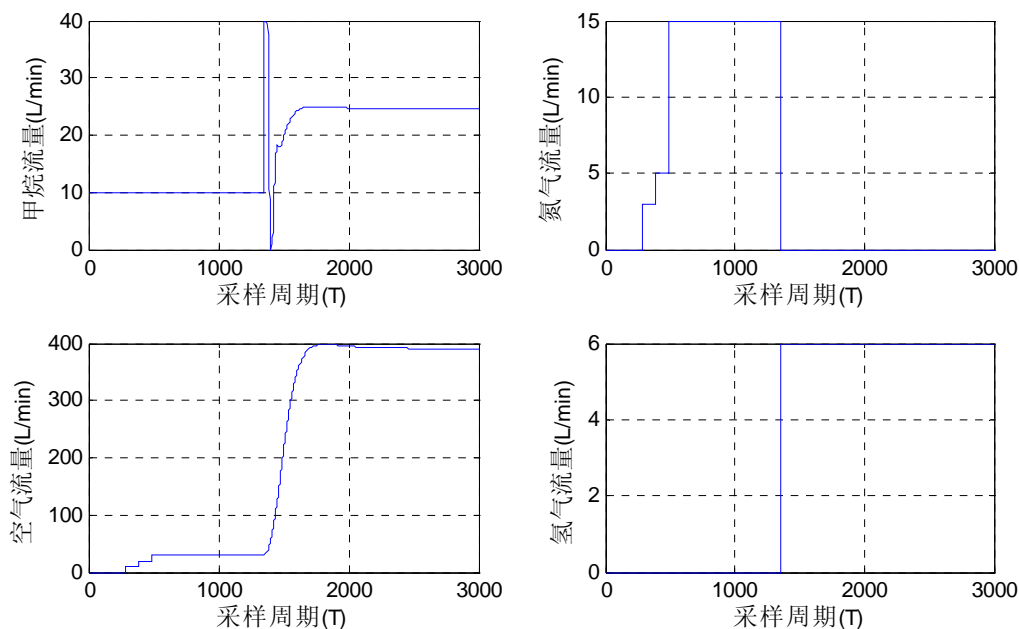


图 3-19 统一后的流量变化曲线图

从图 3-19 的流量变化曲线可以清楚地看到在 1300 个采样周期之前是 StateFlow 逻辑控制器在起作用，氮气的流量从 3L/min 变为 15L/min，空气的流量从 10L/min 变为 30L/min，甲烷的流量和氢气的流量始终不变分别为 10L/min 和 0；在第 1300 个采样周期，电堆入口空气温度到了 650℃，预热阶段结束，SOFC 进入发电阶段，此时 StateFlow 逻辑控制器不起作用了，PI 串级控制器开始发挥作用，甲烷流量到达限幅值 40L/min，氮气的流量变为 0，氢气的流量从 0 变为 6L/min，通入的空气流量大大升高，以带出电化学反应放出的热量；从各气体的流量变化可以清楚地看出系统从 StateFlow 逻辑控制器到 PI 串级控制器的切换过程，所以两种控制器设计和统一是合理的，到此 SOFC 热管理系统的控制器的设计也就完成了。

### 3.4 本章小结

本章主要进行了 SOFC 热管理系统控制策略的研究。SOFC 系统发电过程中有两个阶段：预热阶段和发电阶段，根据两个阶段的不同要求，对预热阶段，进行了 StateFlow 逻辑控制器的设计；对于发电阶段，在上一章建立的系统温度模型的基础

上，首先采用了两个 PI 控制器对电堆入口空气温度和电堆温度进行控制，但是发现控制系统的抗干扰能力不强，可能因此导致电堆因温差过大损坏。通过分析发现系统抗干扰能力不强的原因是系统本身存在较大的迟滞，在此基础上设计了关于电堆入口空气温度的串级 PI 控制器，仿真结果表明它能够很好的抑制扰动，同时也加快了系统的相应速度。最后，为了实现系统从预热阶段到发电阶段的过渡，把 StateFlow 逻辑控制器和串级 PI 控制器合理的统一了起来。

## 4 控制策略在 XPC 平台的实现和验证

上一章对 SOFC 热管理系统的重要组成部分即控制策略进行了深入研究，并且设计了相关控制器，本章将重点探讨 SOFC 热管理系统的第三个重要组成部分即控制策略的实现。传统控制策略实现方式中存在需要手动编程，效率低下，不能把注意力集中在控制策略的研究上而需要投入大量精力在编程方法技巧上的缺陷，在本文的 FameView-PLC 平台上这个问题显得更加突出，因为 PLC 控制器只适合一些简单的控制算法，一旦需要将简单控制算法需要升级成比较复杂的控制算法，那么在 PLC 上进行复杂控制算法的编程和验证将是一件十分头疼的事情，本章将采用快速控制原型技术，通过编写 XPC 实时操作系统下的 PCI 板卡驱动程序，把上一章得到的控制策略转化为 XPC 平台上实时控制的代码，再通过 PCI 硬件设备将控制算法运行的结果送出去控制被控对象，以验证控制策略的有效性；这种 XPC 实时控制平台可以完全替换掉 FameView-PLC 平台，使控制策略的设计变得事半功倍，大大提升开发效率。

### 4.1 快速控制原型与 XPC 平台

快速控制原型（Rapid Control Prototyping - RCP）是实时仿真的一种，实现于产品研发的算法设计阶段与具体实现阶段之间，它通过某种手段将工程师开发的算法下载到某个计算机硬件平台中，该计算机硬件平台在实时条件下运行，模拟控制器，通过实际的 I/O 设备与被控对象实物连接，验证算法的可靠性和准确性。要实现快速控制原型，必须有集成良好、便于使用的建模、设计、离线仿真、实时开发及测试工具，并且允许反复修改模型设计，进行离线及实时仿真。采用 RCP 技术，可以在产品投产之前，方便、快捷的进行控制策略的测试，通过测试就可以仔细研究诸如离散化、采样频率及控制算法的合理性和性能，并对其中存在的问题进行修正，将错误及不当之处消除于设计初期，使设计修改费用降至最小，同时缩短产品的开发

时间。

MATLAB 是一种科学计算软件，专门以矩阵的形式处理数据，具有强大的数值计算和分析能力，并且拥有数据可视化以及易与使用的编程环境等突出优点，因此在科学计算、控制系统、信息处理等领域得到了广泛应用。MATLAB 还针对实时系统的仿真和产品的快速原型化提供了一个特殊应用工具箱 Real-time Workshop(RTW), XPC 就是其中一种基于 RTW 体系框架的辅助产品，它可将奔腾机或 PC 兼容机变成一个实时系统也即 XPC 平台，从而实现快速控制原型或硬件在回路仿真。XPC 平台采用宿主机-目标机的技术途径，及“双机”模式，其中宿主机用于运行 MATLAB/Simulink 进行离线仿真，目标机中运行 XPC 实时操作系统，执行宿主机通过 RTW 生成的 XPC 目标实时代码，通过 I/O 硬件检测被控对象信号和发出控制信号到被控对象进行在线仿真，从而实现快速控制原型；宿主机和目标机通过以太网或串口连线来进行通信。XPC 双机模式的工作原理如图 4-1 所示：

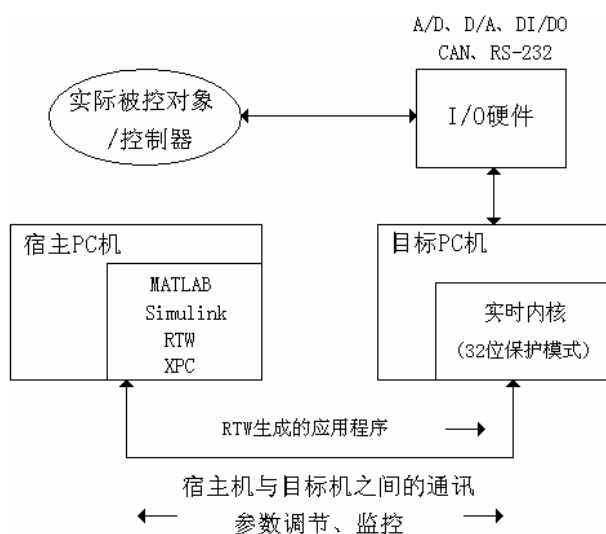


图 4-1 XPC 双机模式的工作原理图

虽然 MATLAB 提供了 XPC 这一有利的实时仿真硬件平台，但是这个平台还不完善，它就像一个断了手脚的废人，无法和外界进行信息交互，为了接上它的手脚需要数据采集卡的支持，提供控制算法和被控对象进行信息交互的硬件基础，但是这还仅仅是接上了手脚，还没有打通它的经脉，仍然是一个残废，由于 XPC 实时操作系统无法识别数据采集卡设备，所以必须还需要编写 XPC 实时操作系统下的数据

采集卡的硬件驱动程序，这样才算是续上了经脉，使 XPC 成为一个完整的实时仿真硬件平台，所以搭建 XPC 硬件平台的关键就是根据需求选择合适的数据采集卡，并编写 XPC 实时操作系统下数据采集卡的硬件驱动程序。下面将以此为重点，介绍 XPC 平台的搭建步骤。

## 4.2 XPC 平台的搭建

为了搭建 XPC 平台，首先必须在目标机上加载 XPC 实时操作系统，因为只有目标机加载了操作系统后才能下载 RTW 生成的实时应用程序。目前，在 MATLAB 帮助文档中仅仅只能查阅到 XPC 目标机的软盘加载方式，但是目前的 PC 机普遍淘汰了软驱，所以使用软盘加载方式会带来极大的不便。为此，必须寻找新的 XPC 目标机实时操作系统加载方式，本文在查阅了其它大量文献的基础上，采用了一种新的 U 盘加载方式。采用 U 盘操作系统加载方式可以从 U 盘加载 XPC 实时操作系统到内存中，同时不会损坏硬盘文件，方便快捷，所以是一种比较理想的加载方式。

XPC 的 U 盘加载方式通过宿主机的 XPC 管理器把 XPC 实时操作系统复制到 U 盘中；目标机启动时插入 U 盘，通过 U 盘把 XPC 实时操作系统加载到目标机的内存中，这种加载方式要通过以下步骤实现：

- 1) 插入 U 盘到宿主机中，把 U 盘设置为 DOS 启动盘，使目标机通过 U 盘启动的方式启动到 DOS 环境下。在本文中采用 USBot1.67 工具将 U 盘设置为 ZIP 方式下的 DOS 启动盘，DOS 启动盘的制作图如图 4-2 所示：



图 4-2 DOS 启动盘的制作图

- 2) 打开 MATLAB，在 XPC 管理器下设置对话框完成对 XPC 实时操作系统文件的

配置：包括对宿主机-目标机通信方式的选择，目标机可使用内存大小的配置以及允许的最大 Simulink 模型大小的配置等。在本文中选择宿主即-目标机的串口通信方式；目标机的内存大小选择为自动，这样可以使目标机所拥有的所有内存；允许的最大模型大小选择为 4M。XPC 管理器的对话框配置图如图 4-3 和图 4-4 所示：

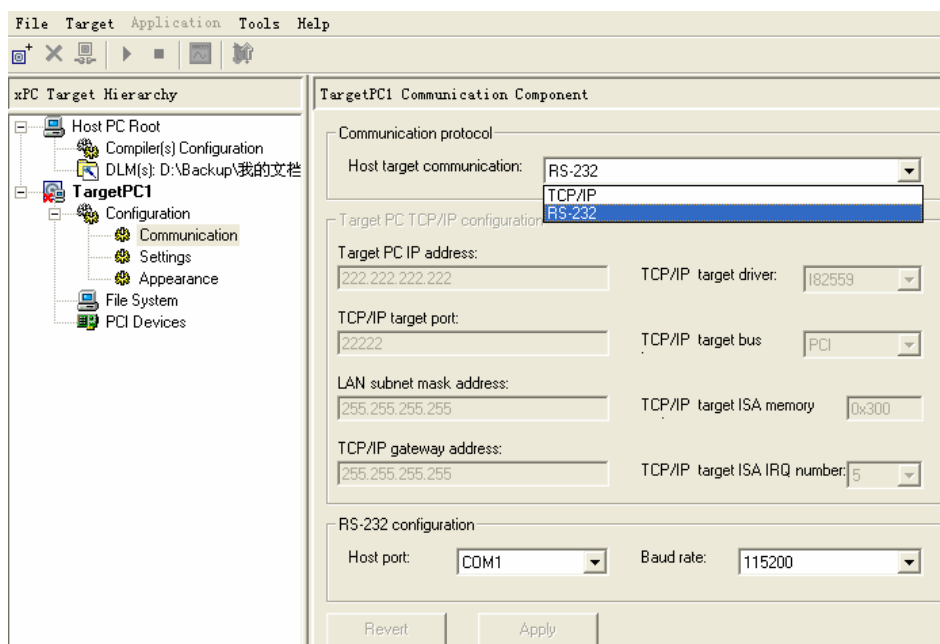


图 4-3 XPC 通信方式配置图

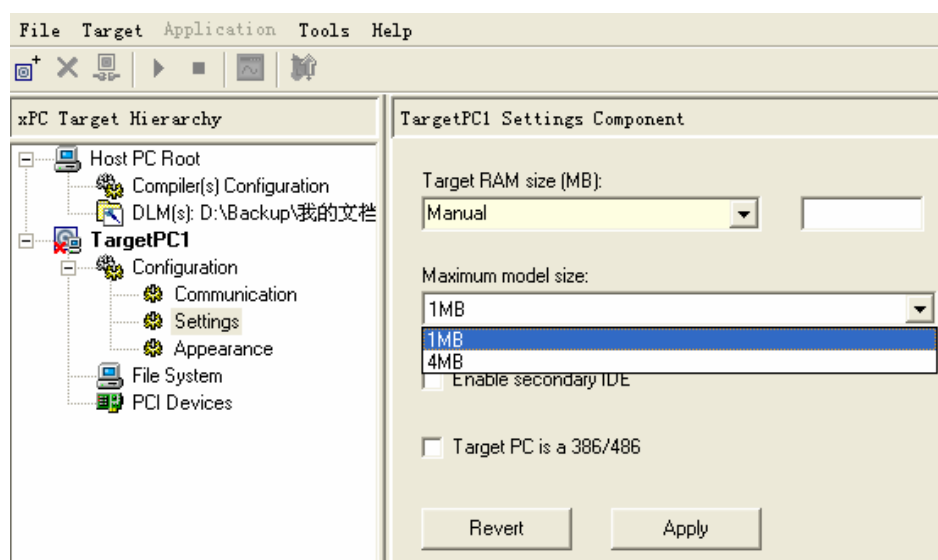


图 4-4 目标机内存及模型大小配置图

3) 生成 DOS 载入器的目标启动盘。首先在 XPC 管理器中的配置对话框中选中 DOSLoader 方式，如图 4-5 所示；然后在管理工具/计算机管理/磁盘管理中把 U 盘的盘符改为 A 盘，因为在 DOSLoader 模式下，XPC 管理器将实时操作系统文件复制到 DOS 启动盘上，而其默认的目标盘路径为 A 盘，所以必须把 U 盘的盘符改成 A 盘，这是 XPC 目标机的 U 盘启动盘能否制作成功的关键性一步，盘符修改操作图如图 4-6 所示；最后点击图 4-5 中的 CreatBootdisk，将 XPC 操作系统复制到 U 盘中。

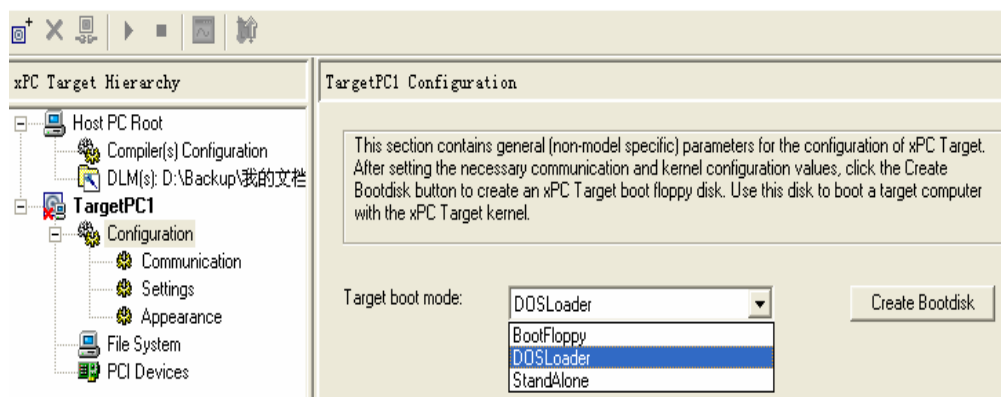


图 4-5 XPC 的 U 盘启动方式对话框配置图

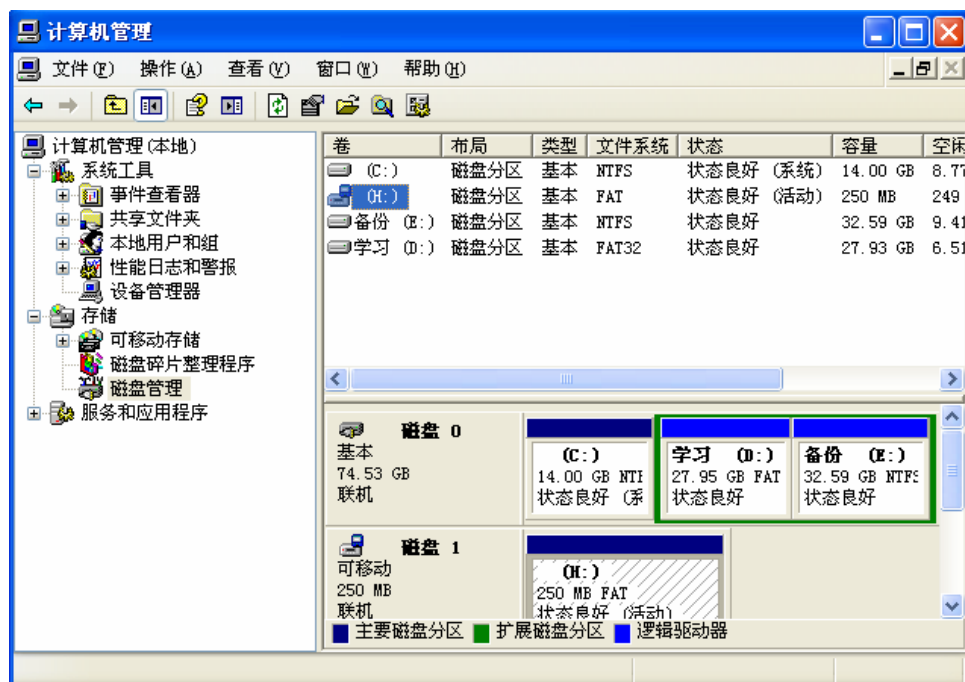


图 4-6 盘符修改操作图

- 4) 把 U 盘从宿主机中拔下来插入目标机并启动目标机，在目标机中设置 U 盘启动为第一启动方式，这样目标机就可以成功加载 XPC 实时操作系统，加载成功后的图如图 4-7 所示：



图 4-7 目标机成功加载 XPC 实时操作系统图

为了搭建 XPC 平台，在目标机成功加载了 XPC 实时操作系统后，需要选择合适的数据采集卡，它起着串联被控对象和目标机的作用。目标机只有通过它才能与实际被控对象进行信息交互，实现实时仿真。PCI-1711 是研华公司生产的一款功能强大的低成本多功能 PCI 总线数据采集卡，完全符合 PCI 规格 Rev2.1 标准，支持即插即用，在安装插卡时，不需要设置任何跳线和 DIP 拨码开关，所有与总线相关的配置，如基地址、中断，均由即插即用功能完成；并且其片上资源十分丰富，完全可以满足 XPC 目标机和被控对象信息交互的需求。

PCI-1711 数据采集卡的片上资源有：1)16 路单端模拟量电压输入通道，每个输入通道的增益可编程，分为  $\pm 10V$ ， $\pm 5V$ ， $\pm 2.5V$ ， $\pm 1.25V$ ， $\pm 0.625V$  等 5 个等级，模拟量输入通道对应 12 位的 A/D 转换器，采样速率可达 100KHz，转换时间为 10us，并且具有 1K 采样 FIFO 缓冲器；2)2 路 12 位模拟量电压输出通道，根据不同的参考电压其输出电压范围也不相同：采用内部参考电压时，其输出电压范围可以设置为 0~5V 或 0~10V；采用外部参考电压时，其输出电压范围为 0~XV，其中 X 是参考电压；3)16 路数字量输入及 16 路数字量输出通道；4)可编程触发器/定时器。PCI-1711 数据采集卡的实物图如图 4-8 所示：



图 4-8 PCI-1711 数据采集卡实物图

为了搭建 XPC 平台，在选择了 PCI-1711 数据采集卡之后，XPC 目标机就具有了与实际被控对象进行信息交互的硬件基础；但是目标机上运行的 XPC 实时操作系统并不能识别出 PCI-1711 数据采集卡，为此必须还要编写 PCI-1711 数据采集卡在 XPC 实时操作系统下的驱动程序，为 PCI-1711 硬件设备提供软件支持。

XPC 操作系统下设备驱动中程序的开发与 Windows 操作系统下驱动程序的开发是不相同的：XPC 系统不是一个完整的实时操作系统，仅仅是一个实时内核，XPC 下的驱动程序可以直接操作设备硬件资源（I/O 端口或内存）；而 Windows 操作系统对硬件是分层操作的，不允许直接操作 I/O 端口或内存。因此，XPC 操作系统下设备驱动程序的开发要相对简单一些。为了实现 XPC 快速控制原型，必须在 Simulink 中调用 XPC 实时内核下的 PCI-1711 数据采集卡的驱动程序，这样才能把实时仿真算法和 I/O 设备结合起来与实际被控对象进行信息交互，实现实时仿真；所以问题就转化为如何在 Simulink 环境下编写基于 XPC 实时内核的 PCI-1711 设备驱动程序，而其关键就在于如何在 Simulink 中访问 PCI-1711 设备的硬件资源。MATLAB 提供了一种基于 C 语言的 S 函数，它可以直接对 PCI-1711 设备的硬件资源进行操作，将它编译后会生成一个动态链接库，再把这个动态链接库和 Simulink 下的 S-Function 模块绑定就可以与其它 Simulink 模块进行信息交互；因此它起着沟通 Simulink 和 PCI-17011 硬件资源的桥梁作用，也就是所要编写的驱动程序。根据这种特性，先编

写基于 C 语言的 S 函数对 PCI-1711 的硬件进行操作，然后通过 Simulink 来调用它，通过这种方式就能实现目标机中 Simulink 模型算法与实际被控对象的信息交互，实现 XPC 快速控制原型。

基于 C 语言的 S 函数有固定的结构，它在 Simulink 下的工作原理图如图 4-9 所示：

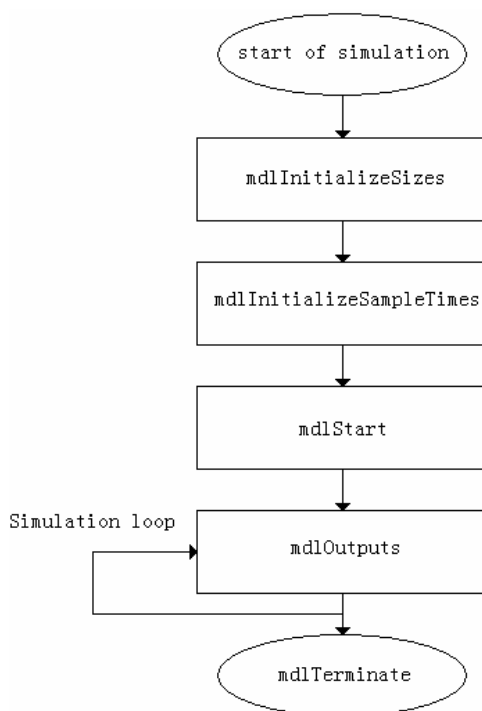


图 4-9 S 函数在 Simulink 下的工作原理图

从图 4-9 看出，基于 C 语言的 S 函数由 `mdlInitializeSizes`，`mdlInitializeSampleTimes`，`mdlStart`，`mdlOutputs` 和 `mdlTerminate` 等 5 个部分组成，它们在 Simulink 仿真时起着不同的作用。`mdlInitializeSizes` 定义它的输入量、输出量、状态量、参数量等，完成对 S 函数与 Simulink 的外部接口和其内部状态的初始化；`mdlInitializeSampleTimes` 完成对 S 函数采样时间的初始化；`mdlStart` 完成对 I/O 硬件设备的初始化；`mdlOutputs` 在仿真运行时对它进行连续回调，每经历一次仿真步长，`mdlOutputs` 就被调用一次，直到仿真结束为止；`mdlInitializeSizes` 起着在仿真结束后把 I/O 硬件置于安全状态的作用。

在 PCI-1711 数据采集卡插入到目标机之后，目标机会在它的 I/O 空间中为它分

配 32 个连续地址，因此为了对 PCI-1711 设备的硬件资源进行操作，必须先找到它在目标机 I/O 空间中对应的基地址，这也是在编写基于 C 的 S 函数的驱动程序中必须解决的难点问题。MATLAB 提供的头文件“pci\_xpcimport.h”中的函数 `rl32eGetPCIInfo()` 可以完成 PCI 设备的查找定位，给出板卡的设备标识 (DID) 和厂商标识 (VID) 并得到 PCI 配置空间信息，其函数定义为：`rl32eGetPCIInfo((unsigned short)Vendor_ID, (unsigned short)Device_ID, &pciinfo)`，它将获得的 PCI 设备信息存储到了 `pciinfo` 结构体中，`pciinfo` 结构体属于 `PCIDeviceInfo` 类型，其定义如下：

```
typedef struct PCIDeviceInfostruct{
    unsigned long BaseAddress[6];
    unsigned short AddressSpaceIndicator[6];
    unsigned short Memory Type[6];
    unsigned short Prefetchable[6];
    unsigned short InterruptLine;
}PCIDeviceInfo;
```

其中基地址信息就包含在数组 `BaseAddress` 中。根据 PCI 协议，PCI BIOS 可给设备分配 6 个基地址，但是其中某些基地址是无效的。为了在 XPC 驱动程序中确定 PCI-1711 有效的基地址，本文在调试程序时通过调用“pci\_xpcimport.h”头文件中的函数 `rl32eShowPCIInfo(pciinfo)`，将 `pciinfo` 的结构体信息在目标机的显示器上显示出来，从而确定 PCI-1711 的基地址存放在 `BaseAddress[2]` 中。这样在驱动程序中就可以通过“基地址+偏移地址”的方式，利用基于 C 语言的 S 函数完成对 PCI-1711 硬件资源的操作。程序编写完毕后将它保存为 C 文件；然后在 MATLAB 中输入 `mex + C 文件名的命令` 后，会生成一个与 C 文件同名的动态链接库文件；接着在 Simulink 中的 S-Function 模块中输入 C 文件名和 S 函数的参数，就可以通过 S-Function 模块来调用驱动程序了；最后进行对 S-Function 模块进行封装，就可以向驱动程序传递 S 函数的参数了。以 D/A 驱动程序为例，封装前 S-Function 的对话框配置如图 4-10 所示，S-Function name 对话框输入的是驱动程序的 C 文件名，S-Function parameters 输入的是驱动程序中的需要赋值的函数参数名；封装后的 S-Function 的参数设置图如图 4-11 所示，Channel vector 输入的是需要进行 D/A 输出的通道号，Range vector 输

入的是 D/A 输出通道的输出范围，Sample Time 输入的是采样时间。

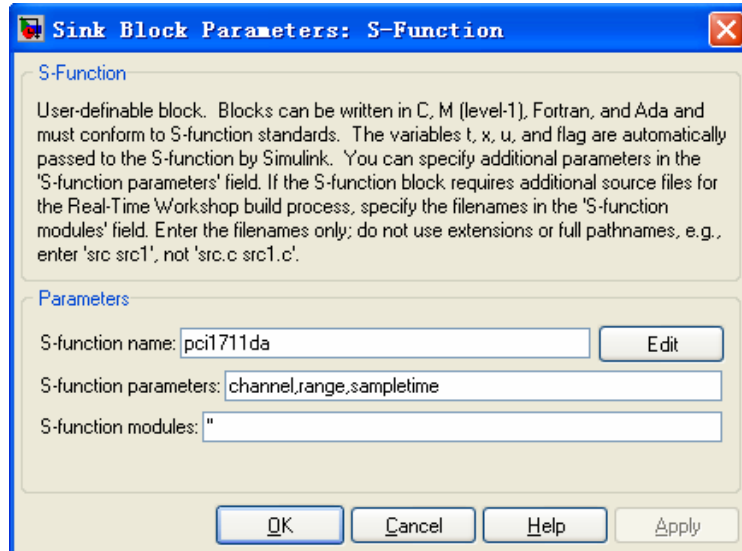


图 4-10 封装前 S-Function 的对话框配置图

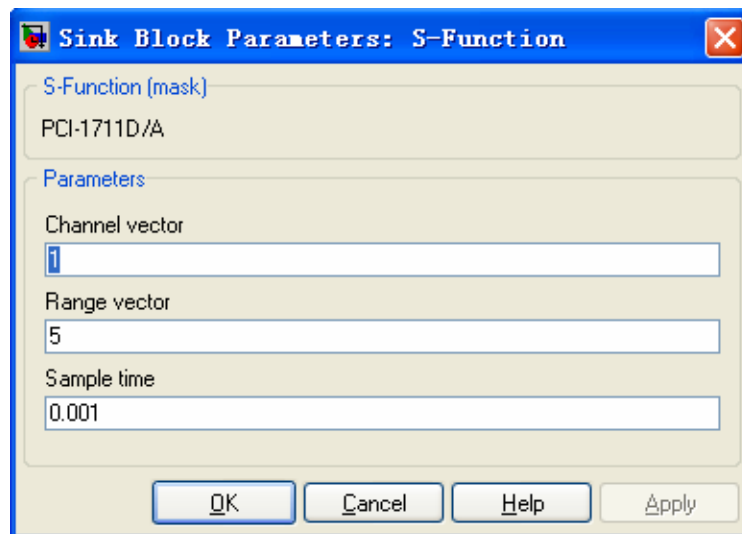


图 4-11 封装后 S-Function 的参数设置图

为了验证驱动程序的正确性，设计了如图 4-12 所示的 Simulink 框图：

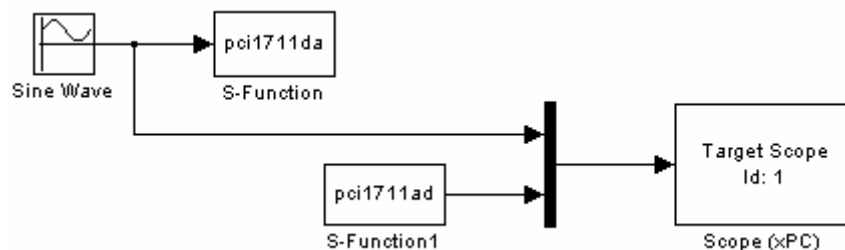


图 4-12 驱动程序的验证图

pci1711da 模块的作用是接受 Simulink 中 Sine Wave 模块的输出值，按照这个值驱动 PCI-1711 数据采集卡进行 D/A 输出，由于它不需要进行 Simulink 输出，所以它没有输出端口；pci1711ad 模块的作用是进行 A/D 转换，把转换后获得的数值在 Simulink 中输出，由于它不需要接受 Simulink 输入，所以它没有输入端口；pci1711da 模块和 pci1711ad 模块对应的 PCI-1711 板卡的 D/A 输出通道和 A/D 输入通道通过实际导线进行连接，让板卡的 D/A 输出成为 A/D 的输入，然后把 A/D 转换结果送给 Target Scope 模块与 D/A 模块给定值进行比较，在目标机上进行一个闭环检测。

为了实现闭环测试，在建立好了 Simulink 测试模型后，点击 Simulation/Configuration Parameters 进行 Solver 和 Real-Time Workshop 的配置。由于 XPC 实时内核不支持变步长，所以 Solver 的仿真步长必须配置为定步长；Real-Time Workshop 的 System target file 设置为 xpctarget.tlc，以使 Simulink 通过 RTW 产生的文件为 XPC 目标文件。配置完毕后点击 Tools/Real-Time Workshop/Bulid Model，生成一个关于这个 Simulink 模型的 XPC 目标文件，其后缀名为 dlm，最后在宿主机的 XPC 管理器中将这个文件下到目标机中执行，就实现了闭环测试。D/A 输出在示波器上的显示结果如图 4-13 所示，比较结果如图 4-14 所示：

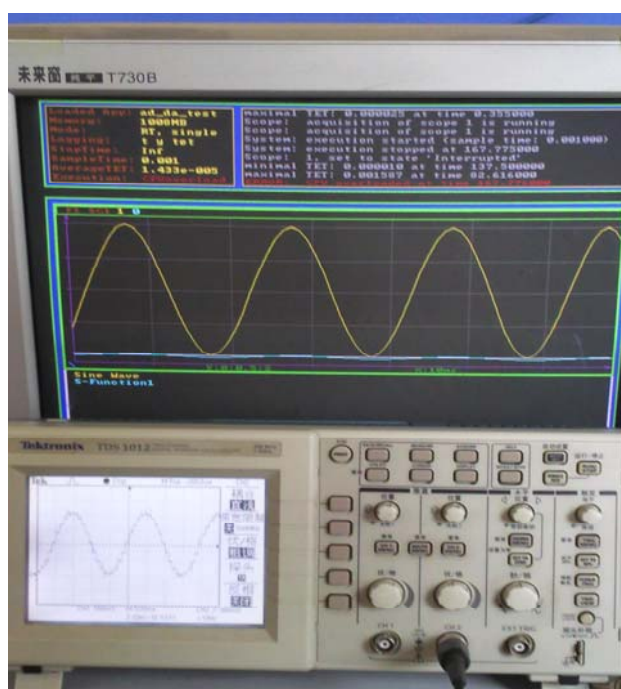


图 4-13 D/A 输出在示波器上的显示结果

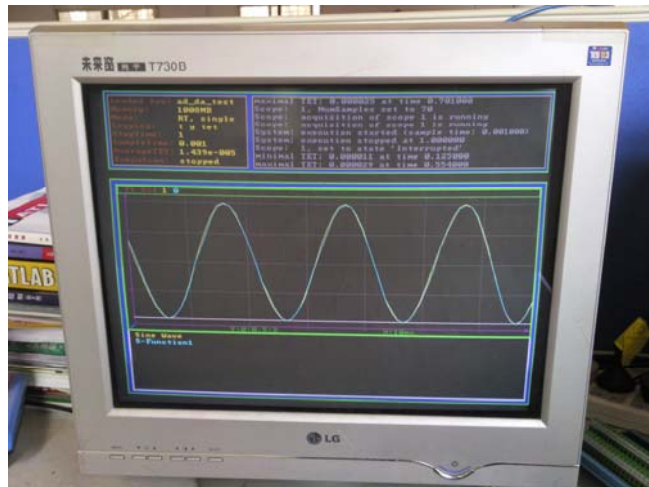


图 4-14 D/A 给定值与 A/D 获得值的比较结果

从图 4-13 和图 4-14 可以看出：基于 C 语言的 S 函数的驱动程序编写合理，能够正常驱动 PCI-1711 的硬件资源。

在驱动程序编写成功后，一个完整的 XPC 硬件平台也就完成了，在此基础上就能很容易的实现 XPC 快速控制原型了，只需要把上一章得到的 Simulink 下 SOFC 热管理系统的控制部分和 S 函数驱动模块相连组成一个 Simulink 模型如图 4-15 所示，然后按照驱动程序测试过程类似的步骤，就可以将这整个 Simulink 模型通过 RTW 生成 XPC 目标文件下载到目标机上，这样就使上一章设计的控制策略在 XPC 目标机上得到了实现。

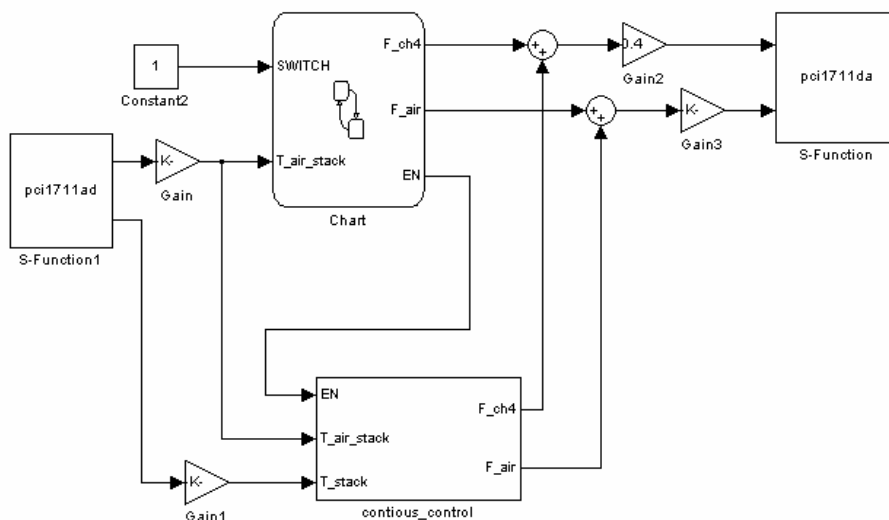


图 4-15 XPC 快速控制原型的 Simulink 结构图

### 4.3 硬件在回路仿真与控制策略验证

由于 SOFC 热管理系统的温度控制试验费用开销较大，并且十分耗时，为此在本文中采用硬件在回路仿真技术，利用 XPC 这个实时仿真平台来运行第二章中辨识出的被控对象模型通过 RTW 后生成的 XPC 目标代码对实际的 SOFC 热管理系统的被控对象进行模拟，然后在这个 XPC 平台上进行试验，这样可以大大减少试验成本和缩短试验时间。

为了对上一章得到的 SOFC 热管理系统的控制策略进行验证，在本文中利用一个 XPC 实时仿真平台来实现快速控制原型，使其成为控制器；利用另一个 XPC 实时仿真平台来实现硬件在回路仿真，对实际的被控对象进行模拟；两个 XPC 平台上的 PCI-1711 数据采集卡通过导线进行互连，实现控制量信息和测量量信息的交互；通过这种方式来进行控制策略的验证。XPC 实时仿真平台下的控制策略验证结果如图 4-16 和图 4-17 所示：



图 4-16 被控量的在 XPC 上的显示结果



图 4-17 控制量在 XPC 上的显示结果

从图 4-16 的被控量的控制结果看出电堆温度被控制在了  $750^{\circ}\text{C}$ ，电堆入口空气温度被控制在了  $720^{\circ}\text{C}$ ；从图 4-17 的气体流量控制量可以看出系统从逻辑控制到连续控制发生了正常切换；综合图 4-16 和图 4-17 可以看到第 3 章设计的控制策略在 XPC 平台上得以成功实现，从而证明控制策略的设计是合理的，可以很好的完成对 SOFC 热管理系统电堆入口空气温度和电堆温度的控制。

#### 4.4 本章小结

本章为了验证上一章控制策略的合理性，针对传统控制策略实现方式中需要手动编程，效率低下的缺陷和 SOFC 试验费用开销较大，十分耗时的问题分别采样了基于 XPC 实时仿真平台的快速控制原型和硬件在回路仿真技术，将一个 XPC 平台作为控制器，实现了上一章的控制策略；另一个 XPC 平台模拟 SOFC 系统被控对象；通过这种方式验证了控制策略的合理性，能很好的满足控制要求。

## 5 UCOS 实时操作系统在 MPC555 上的移植

上一章介绍了通过 XPC 快速控制原型的方式来实现 SOFC 热管理系统的控制策略，这种控制策略的实现方式方便、快捷，能够充分调动控制策略设计者的积极性，是一种十分高效的开发方式；但是 XPC 平台是以 PC 机作为控制器，体积太过庞大，在实际应用中会受到很过限制，只能作为过渡期的控制器平台，为了向产品级 SOFC 热管理系统的方向努力，必须提高系统的集成度，因此替换掉体积庞大的 XPC 控制器是必然的。在 MATLAB 的 RTW 工具箱中支持 MPC555 微处理器下的快速控制原型实现，采用 MPC555 处理器和其外围硬件系统来替代 XPC 平台，就可以在缩小了控制器体积的同时还保留快速控制原型的实现方式，因此用 MPC555 硬件平台来作为最终产品级 SOFC 热管理系统的控制器十分合适。

### 5.1 MPC555 硬件平台简介

本文中的 MPC555 硬件平台是从美国 PHYTEC 公司购买的一套基于 MPC555 芯片及其外围硬件扩展系统的开发板，它由核心板和 I/O 接口板组合而成，核心板实物结构图如图 5-1 所示，I/O 接口板实物结构图如图 5-2 所示：

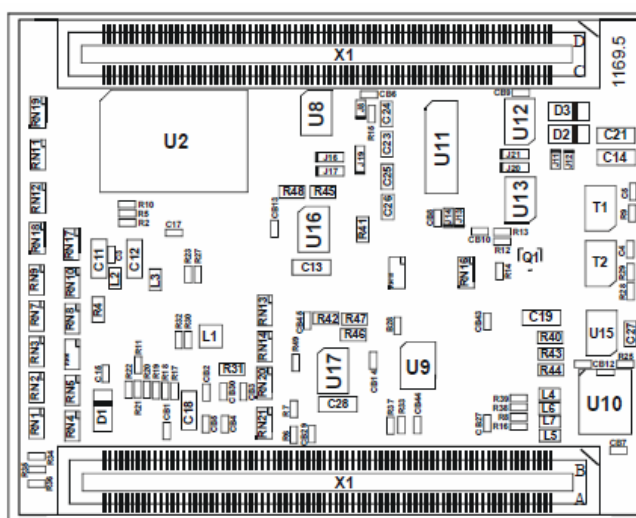


图 5-1 核心板实物结构图

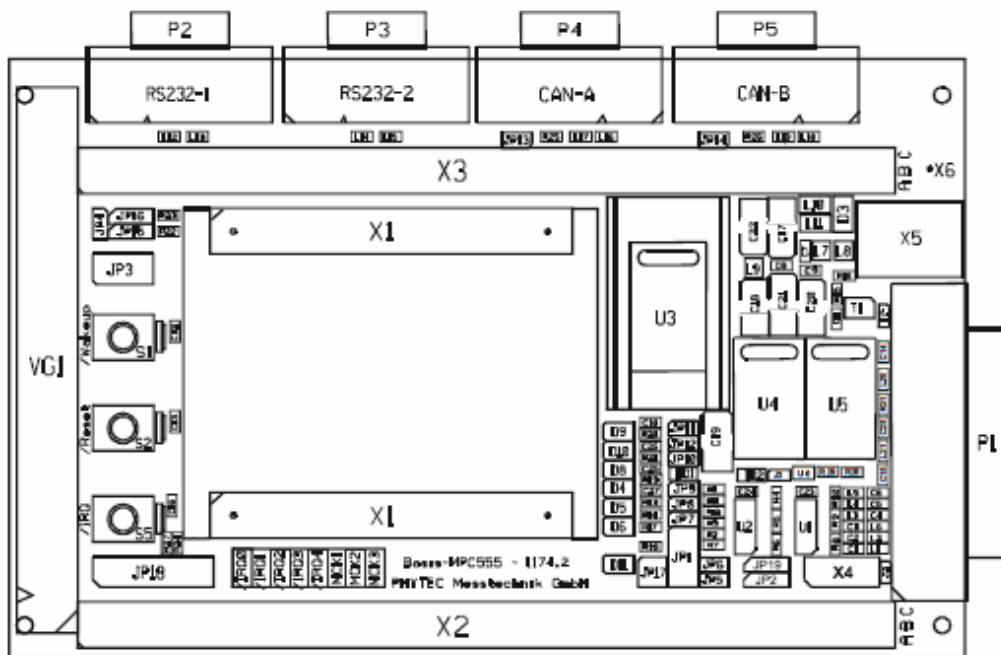


图 5-2 I/O 接口板实物结构图

核心板由 Motorola 公司生产的 PowerPC 精简指令级微处理器 MPC555 和 1M 的片外 RAM 与 4M 的片外 Flash 组成，其中 MPC555 微处理器具有处理速度快，片上资源丰富的突出优点：1)具有高达 40Mhz 的系统处理速度 2)具有独立的浮点计算单元，3)包含 26K 的快速 RAM 和 448K 可编程的 Flash，4)具有队列串行多通道模块和两个 CAN2.0B 控制器模块，5)两个 TPU 时钟定时器模块，6) 2 组共 32 路模拟量输入通道的 A/D 转换器。

I/O 接口板是核心板的外围硬件电路扩展，它由两个 RS-232 串口、两个 CAN 通讯口、以及 VG1 上的 96 个 I/O 端口组成，它是核心板和外部信号进行交换的中介。

由于目前的 MPC555 平台是一款通用的开发板，针对性不强，为了课题的需要，必须对它的软硬件进行改进和拓展。在硬件上，由于 MPC555 硬件平台没有 D/A 驱动电路，并且它的 I/O 接口板只提供了外部 I/O 端口，并未真正实现硬件电路拓展，所以必须在这些方面进行改进和完善；在软件上，目前 MPC555 平台只实现了快速控制原型，但是采用 MPC555 平台来替代 XPC 平台的最终目的是让其作为产品级的控制器，因此它不但要实现快速控制原型，而且还要具备通讯、监控、参数配置、中断等其它相关的功能，但是在加入这些额外功能后无疑会拖累控制算法的执行速

度，可能会导致控制算法失效，为了在加入这些额外功能后，保证控制算法的实时性和有效性，必须引入实时操作系统，将控制算法设为优先级最高的任务，其它额外功能设置为低优先级任务，然后再由操作系统来对它们进行调度，其中实时操作系统的选择和移植将是关键。由于本文讨论的重点不是对 MPC555 硬件的改进和完善，而是对其软件部分进行实时操作系统的移植，因此下面仅对这方面进行详细研究和说明。

## 5.2 UCOS 实时操作系统简介

实时操作系统是指如果逻辑和时序出现偏差将会引起严重后果的系统。通常可将实时操作系统分为两大类，软实时系统和硬实时系统。在软实时操作系统中要求各任务越快执行越好，但是没有限定时间界限，要求在多长时间内完成任务；在硬实时操作系统中，各任务不仅要执行无误而且要准时，现在大多数实时操作系统是它们的结合。与实时操作系统系统相对的是前后台系统，前后台系统的后台一般是应用程序的无限循环，通过调用相关的函数来实现某些特定功能；前后台系统的前台是中断服务程序，通过它来对一些马上需要处理突发事件进行操作。与前后台系统相比，实时操作系统能够充分利用资源，对多个任务进行操作，具有很强的实时性。

目前，实时操作系统的种类很多，比较有代表性的是 Linux, WinCE, VxWorks, UCOS 等，它们在食品加工、化工厂等过程控制方面；在发动机控制、防抱死系统等汽车行业方面；在飞机管理系统、武器系统、喷气发动机控制等航空航天方面都得到了广泛应用。由于 WinCE、VxWorks 等实时操作系统都是商业操作系统，其源代码并不完全公开；Linux 虽然源代码公开，但是其内核文件太大，并且移植起来并不容易；而 UCOS 是一种免费公开源代码、结构小巧、具有可剥夺实时内核的实时操作系统，它具有执行效率高、占用空间小、实时性能优良和可扩展性强，移植比较方便等突出优点；所以本文采用 UCOS 实时操作系统在 MPC555 平台上进行移植，以提高控制算法的实时性。

### 5.3 UCOS 操作系统的内核结构

为了在 MPC555 硬件平台上进行 UCOS 实时操作系统移植，首先必须了解其内核结构，熟悉其工作原理。UCOS 操作系统的内核结构包括：任务和任务控制块、任务就绪表和任务调度、时钟节拍等几个重要方面，下面将对它们进行重点介绍。

在 UCOS 实时操作系统中，任务是操作系统进行调度和运行的基本单位，每个任务在操作系统中都相当与前后台系统中的一个后台，由一个无限循环构成。一个复杂的应用程序，可以把它分解成若干个简单的任务，然后让操作系统再对这些任务进行调度，让它们协调完成这个复杂的任务。UCOS 操作系统具有很强的实时性的关键因素是系统会进行任务调度，在任务调度发生时，与调度紧密相关的两个任务会从一种状态变成其它状态。UCOS 中存在 5 种任务状态，系统中的每个任务都只可能处于这 5 种状态中的一种，任务可以在这几种任务状态之间进行切换，这 5 种状态如图 5-3 所示：

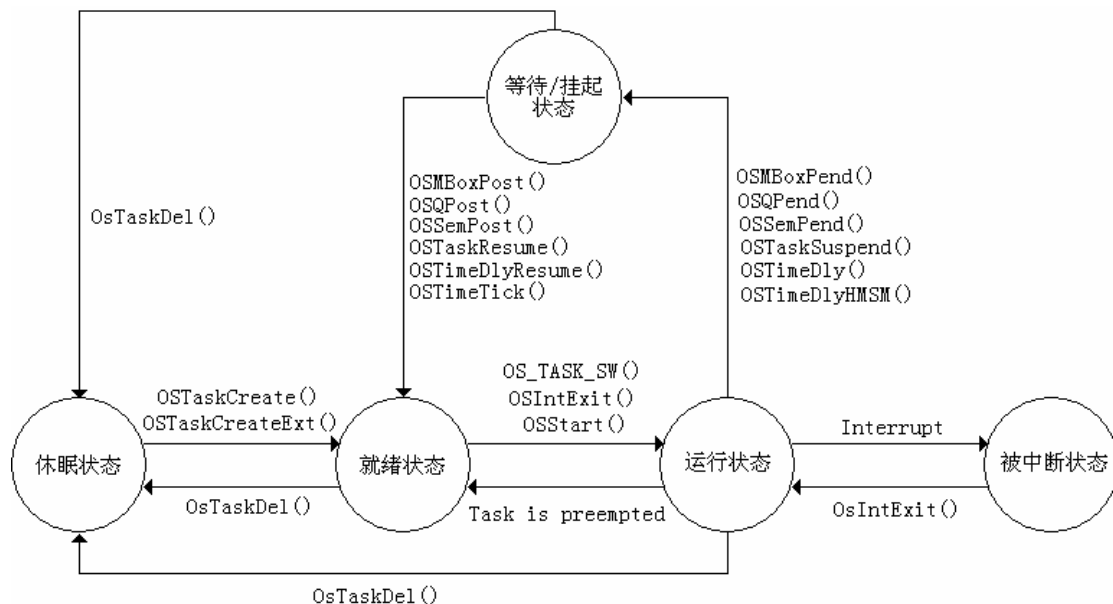


图 5-3 UCOS 系统状态变迁图

为了对任务进行管理，在每个任务建立的时候，UCOS 系统都为它都会创建自己的任务堆栈和任务控制块，任务堆栈用来存放 CPU 寄存器的值，而任务控制块用来存放任务的相关信息包括任务堆栈的栈顶指针、任务的状态字、任务的优先级

等。当发生任务调度时，任务控制块会记录下当前的任务信息，等到这个任务下次再执行的时候再根据这些记录信息继续执行程序，这样就可以保证任务在被切换后再次执行时，能够丝毫不差的从程序的切换点继续执行。

UCOS 操作系统的核心工作机制是找出处于就绪状态中最高优先级的任务，让其获得 CPU 的控制权使其优先运行。在 UCOS 操作系统中，系统可以按照任务的不同重要程度分配最多 64 个任务优先级，并且每个任务的优先级都不能相同。为了快速找到处于就绪状态的最高优先级任务，在 UCOS 系统中采用了任务就绪表来记录就绪状态的任务，它由变量 OSRdyGrp 和数组 OSRdyTbl[]组成，其工作原理如图 5-4 所示：

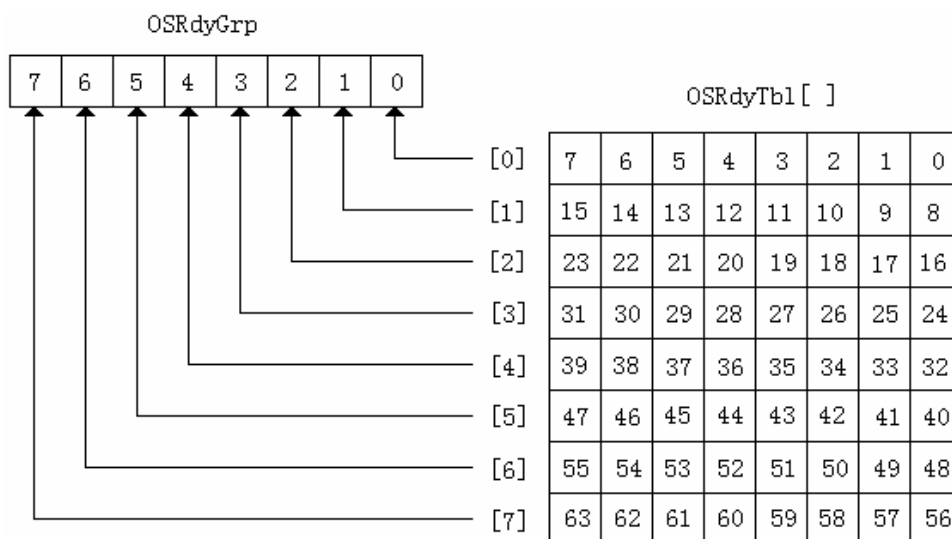


图 5-4 任务就绪表的工作原理图

数组 OSRdyTbl[] 将 64 个优先级任务分成了 8 组，数组元素的每一位记录了不同优先级任务的就绪状态：任务就绪就在相应位标记 1，任务没有就绪就在相应位标记 0；变量 OSRdyGrp 记录了数组 OSRdyTbl[] 中每组元素是否有任务处于就绪状态，如果有则在相应位标记 1，如果没有则在相应位标记 0。然后通过：

```

y = OSUnMapTbl[OSRdyGrp];
x = OSUnMapTbl[OSRdyTbl[y]];
prio = (y << 3) + x;

```

这三条命令就可以在任务就绪表中快速找到处于就绪状态任务的最高优先级，其中 OSUnMapTbl 是一个包含 256 个元素的数组查找表，它对应一个字节的数值的所有数值的

最低位。由于在 UCOS 操作系统中每个任务的优先级是唯一的，到了任务优先级也就找到了相应的任务。

在任务就绪表中找到最高优先级的任务后，UCOS 系统就会尝试进行任务调度。系统会首先比较当前运行的任务的优先级是否比任务就绪表中的最高优先级的任务的优先级低，如果不是，那么任务调度不会发生，当前的任务继续运行；如果是，那么任务调度发生，当前任务交出 CPU 的控制权停止运行，在这个过程中它会把 CPU 中的当前寄存器的值保存到自己的任务堆栈中，然后把堆栈的栈顶指针保存在任务控制块中，为再次执行程序作准备，同时处于就绪态的最高优先级任务获得 CPU 控制权，运行自己的任务程序，在这个过程中它将按照任务控制块中的堆栈指针找到任务自己的堆栈，然后把堆栈中的值弹入到 CPU 相应的寄存器中，使 CPU 根据当前寄存器的状态来运行程序。

在 UCOS 操作系统中，要实现任务调度，不可能总是寻找最高优先级的就绪任务，这样的话它会一直占用 CPU，使任务不能执行。为此 UCOS 操作系统利用周期性的中断，只有在发生周期中断时刻才寻找处于任务就绪表中最高优先级的任务，而在其它时刻运行任务程序，这个周期性的中断也称为 UCOS 系统的时钟节拍。时钟节拍率不能太大，如果太大，实际上操作系统就是主要在于寻找最高优先级任务的活，同样任务不能有效执行，一般时钟节拍率在 10-100Hz 之间。同时利用时钟节拍，可以实现任务的延时功能。

由于在每个时钟节拍发生的时刻，可能会发生任务调度，所以必须先要在时钟节拍中断程序中将 CPU 中的的相关寄存器保存在当前被中断任务的任务堆栈中，以免被中断的任务相关信息丢失；在时钟节拍中断程序的最后，在从当前或就绪任务表中最高优先级任务的任务堆栈中把 CPU 寄存器的值恢复出来。

## 5.4 UCOS 系统的移植工作的展开

通过上一节的介绍，可以看出：1)UCOS 操作系统的核心工作原理就是通过任务调度，使任务就绪表中最高优先级的任务优先运行，2)任务调度的实质是需要被终止运行的任务通过入栈操作保留当前 CPU 的运行状态到自己的任务堆栈中，为下次再

次执行做准备；同时将要开始运行的任务通过出栈操作把自己的任务堆栈中的任务运行信息恢复到 CPU 相关寄存器中，让 CPU 根据当前寄存器的值来运行任务程序。以上两点是进行 UCOS 移植的核心指导思想，下面将根据这个指导思想，进行 UCOS 操作系统移植。

在进行 UCOS 系统移植之前需要对 UCOS 系统的软硬件体系结构有所了解，以便知道在对 UCOS 系统移植时需要做哪些工作，从而方便工作的展开。UCOS 系统的软硬件体系结构图如图 5-5 所示：

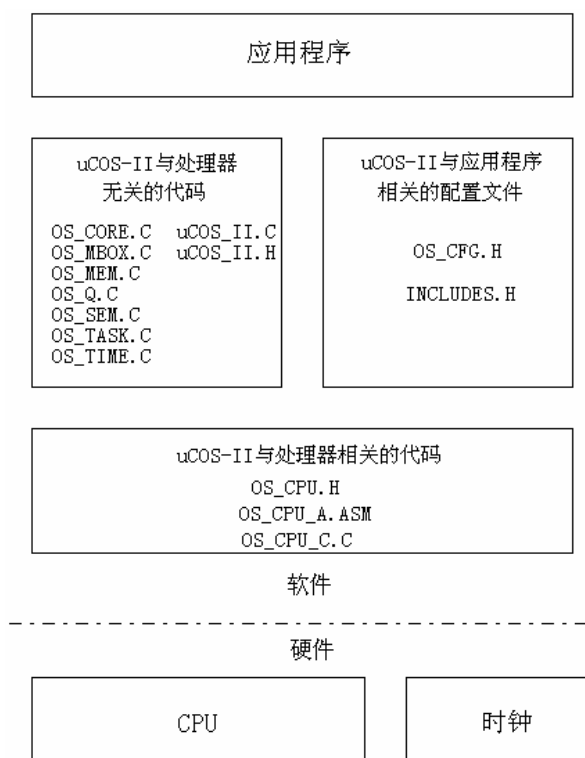


图 5-5 UCOS 系统软硬件体系结构图

从图 5-5 可以看出一个 UCOS 应用系统由三层构成，它们分别是应用层，UCOS 系统层，硬件层；其中 UCOS 系统层处于中间层，它是应用层和硬件层连接的桥梁，它可以对硬件层的硬件进行操作，同时向应用程序提供服务；这样应用层中的应用程序就可以通过调用 UCOS 系统层为了其提供的服务来进行硬件操作和其它操作，不需要关心具体的硬件操作行为，这样程序设计就变得更方便、快捷了。为了向应用程序提供服务，UCOS 系统中提供了与处理器无关的内核调度和相关服务文件以及与应用程序相关的配置文件，其中内核调度和相关服务文件都是固定的不需要改变，

配置文件让程序设计者根据应用程序的具体要求对 UCOS 系统进行配置，选择系统需提供的服务和时钟节拍率等；为了对硬件层中的 CPU、时钟等硬件资源进行操作，UCOS 系统提供了 OS\_CPU.H, OS\_CPU\_A.ASM, OS\_CPU.C 等 3 个与处理器相关的文件，但是这些文件都是不完整的，需要设计者根据具体的处理器进行完善，这是 UCOS 系统移植的工作重点。

#### 1) OS\_CPU.H 文件的完善

在 OS\_CPU.H 文件里，首先要在宏定义中来定义 short,int 和 long 等数据类型，不直接使用它们的原因是它们都是和具体的编译器相关的，编译器不同它们代表的意义也就不一样，这样就可以使设计者按照自己编译器的情况来定义它们，从而有利于 UCOS 系统移植；在定义了系统数据类型之后还要通过编写宏定义函数 OS\_ENTER\_CRITICAL()和 OS\_EXIT\_CRITICAL()来实现开关中断；最后要指定堆栈的增长方向，因为不同的微处理器可能具有不同的堆栈增长方向，从而导致栈顶的指针会不一样。

#### 2) OS\_CPU\_A.ASM 文件的完善

OS\_CPU\_A.ASM 文件是采用汇编语言编写的文件，它里面包含 4 个汇编函数，这些函数都只有定义，需要移植者根据自己的微处理器来进行函数实现，它们分别是 OSStartHighRdy(), OSCtxSw(), OSIntCtxSw(), OSTickISR()。

OSStartHighRdy()函数的作用是在 UCOS 系统刚开始启动任务时，由于此时没有任务运行，需要找到就绪任务表中最高优先级的任务，把它的任务堆栈中的值恢复到 CPU 对应寄存器中，使它开始运行；OSCtxSw()的作用是在当前任务主动放弃 CPU 控制权时，进行任务级的任务切换，它包含的动作有保存 CPU 寄存器的值到当前任务对应的任务堆栈中，再把堆栈指针送给当前任务的任务控制块保存起来，最后找到处于就绪任务表中最高优先级的任务，然后从它任务堆栈中的值恢复到 CPU 对应寄存器中，从而实现任务级切换；OSIntCtxSw()的作用是在中断服务程序结束后，实现中断级的任务切换，它与任务级切换的不同在于，由于在中断服务程序开始时，CPU 寄存器的值已经保存到当前任务的任务堆栈中，所以它的动作只需要把就绪任务表中最高优先级任务的任务堆栈中的值恢复到 CPU 对应寄存器中就可以

了,因此可以看成任务级切换中的一部分;OSTickISR()的作用是进行时钟节拍中断,它包含的动作有保存 CPU 寄存器的值到当前任务的任务堆栈中,执行中断服务,在中断程序结束前判断是否进行任务切换,如果进行切换则调用 OSIntCtxSw()函数,把任务就绪表中最高优先级任务的任务堆栈中的值恢复到 CPU 对应寄存器中,否则,恢复当前任务堆栈的值到 CPU 对应寄存器中。

从这 4 个函数的作用可以看出它们主要是完成任务堆栈和 CPU 相关寄存器的数值交换,由于通过 C 语言无法访问 CPU 的寄存器,这也就是为什么这 4 个函数要用汇编语言来编写,包含在汇编文件中的原因。

### 3) OS\_CPU.C 文件的完善

在 OS\_CPU.C 文件里,主要工作是对任务的堆栈进行堆栈初始化,使任务堆栈看起来就像刚发生过中断,所有与任务运行相关的 CPU 寄存器的值都保存在堆栈中一样,这时堆栈中存放的值是任务运行的初始状态值,确保任务运行有一个正确的初始状态。

在完善了这 3 个与 MPC555 微处理器相关的文件后,移植工作基本上就完成了,接下来就要验证移植工作的正确性,也就是检验这 3 个文件的程序编写是否正确。

由于 UCOS 操作系统的特点是通过任务调度使任务就绪表中最高优先级的任务优先执行,因此可以创建 2 个优先级不同的任务,看它们能否正常进行任务切换来检验移植程序编写的正确性。在 MPC555 平台上通过创建关于 2 个 LED 灯的显示任务,一个高优先级任务是红色 LED 灯以间隔 1 秒的时间进行频闪,另一个低优先级任务是绿色 LED 灯以间隔 2 秒的时间进行频闪,然后让 UCOS 系统开始启动任务,最终发现红色 LED 以 1 秒的时间间隔频闪的同时绿色 LED 灯能够以 2 秒的时间间隔发生频闪,由于两个任务内部都是无限循环,这个现象说明它们能够进行正常的任务切换,从而证明了移植程序编写的正确性。

## 5.5 本章小结

本章从 SOFC 热管理系统控制器平台小型化、产品化的角度考虑,采用体积更小的 MPC555 平台来取代 XPC 平台。作为产品级控制器的 MPC555 平台在应用中不

但要实现快速控制原型，而且要实现其它功能，这样必然会拖累控制算法的运行速度，为了克服这个缺陷，本章在 MPC555 平台上进行了 UCOS 实时操作系统的移植，通过运行 UCOS 系统来提高 CPU 利用率以保证控制算法的实时性。由于目前 MPC555 开发板的硬件结构还不完整，配套的通讯、监控、参数配置、中断等相关的功能都不能有效开展，所以不能进行包括控制算法任务和其它拓展功能任务的多任务实时性测试，本章所做的工作只是一个前瞻性的研究。

## 6 总结与展望

为了将 SOFC 脱离试验台研究应用于实际, 保证其安全、高效的运行, 关键在于为它提供一个稳定的环境温度, 让其工作在最佳工作温度点上, 为此必须为它配备外围热管理系统, 进行外围热管理系统的研究。SOFC 外围热管理系统涉及的方面很多, 主要包括工艺结构设计, 控制算法设计, 控制算法的实现几个重要方面, 其中工艺结构设计是基础, 控制算法设计是方式和手段, 控制算法的实现是最终的落脚点。

在充分借鉴了国内外 SOFC 外围热管理系统的工艺结构设计的基础上, 课题组对其中那些不合理的地方进行的全面改进, 设计出了一套合理的热管理系统工艺结构, 并成功集成了 2.5kw 的 SOFC 外围热管理系统的实物。在此基础上, 本文主要对 SOFC 热管理系统的另外 2 个重要方面控制算法的设计和实现进行了重点研究, 以实现 SOFC 热管理系统的电堆入口空气温度和电堆温度有效控制, 主要完成的研究工作如下:

(1)在对 SOFC 外围热管理系统的工艺结构进行了充分分析和研究的基础上, 结合试验数据, 利用 T-S 模糊模型的建模方法, 在 Simulink 下对工艺结构的各组成部分分别建立了 T-S 辨识模型, 并将它们进行模型集成, 为控制算法的设计打下了牢靠的基础。

(2)SOFC 系统的在发电过程中要经历 2 个阶段, 预热阶段和发电阶段。在预热阶段, 本文设计了 Simulink 下的 StateFlow 逻辑控制器来进行逻辑控制, 这种逻辑控制方式具有层次性好, 设计方便等突出优点; 在发电阶段, 先采用 PI 控制方式对建立的 T-S 辨识模型进行电堆入口温度和电堆温度的控制, 发现这种控制方式存在抗干扰能力不强的缺陷, 通过分析发现这是由于系统的大时滞特性引起的, 为了克服这个缺陷, 本文在 PI 控制的基础上采用串级控制结构, 从而很好的解决了这个问题, 实现了对电堆入口空气温度和电堆温度的有效控制。

(3)本文搭建了 XPC 平台,在该平台下采用快速控制原型的方式来进行控制策略的实现。通过这种方式,将 Simulink 下的控制算法通过自动代码生成技术转化为能够在 XPC 平台上进行实时控制的代码,省去了手工编写代码的烦恼,同时缩短了开发周期。同时,由于做 SOFC 试验的开销比较大,同时试验过程极为耗时,为此在 XPC 平台上采用硬件在回路仿真技术将 SOFC 热管理系统的 T-S 辨识模型通过自动代码生成技术下载到另一个 XPC 平台上,对被控对象进行模拟。通过 2 个 XPC 平台的实时仿真,验证了控制算法的有效性。

(4)由于 XPC 平台体积庞大,在 SOFC 热管理系统应用中会受到诸多限制,只能作为研究阶段的过渡平台,为此本文采用了体积更小的 MPC555 平台来取代 XPC 平台。MPC555 平台作为产品级控制器,除了实现快速控制原型外还要实现其它功能,为了在加入其它功能后不会拖累控制算法的执行速度而导致控制算法失效,本文在 MPC555 平台上进行了 UCOS 实时操作系统移植,从而保证控制算法实时性和有效性。

本文在 SOFC 热管理系统的研究上做了大量有意义和前瞻性的工作,但是由于这个工作的复杂性和时间限制等因素,做的还不够完善,下面这些工作需要进一步的进行深入研究:

(1)本文搭建的 T-S 模糊模型仅一个在负载电流恒定时的热模型,但是在实际应用中负载电流可能是时刻变化的,从而对电堆的温度造成影响,所以还需要考虑负载电流对电堆温度的影响,对热模型进行进一步完善,让其更加贴近现实情况。

(2)由于时间及工作整体性和连贯性的限制,本文并没有对串级控制结构下的控制策略进行深究,只采用了常规 PI 控制方式,是否存在其它高级控制方式有更好的控制效果仍然值得深入。

(3)本文仅仅是对整个 SOFC 系统中的热管理子系统进行了单独研究,但在实际中,必须同时考虑电管理子系统,这两个子系统是密切相关而又相互影响的,热管理子系统仅是基础,电管理子系统才是最终的目的,实现对外部变化负载的跟踪;将这两个子系统合理的统一起来仍是一个艰巨的任务。

(4)MPC555 平台的软硬件经过完善和拓展以后,把它们统一起来仍然还有一些

工作需要完成，如 D/A 驱动程序的编写，同时与之配套的通讯、监控、参数配置、中断等相关的功能的实现，还有在 UCOS 操作系统中，控制算法任务和通讯、监控、参数配置等其它任务如何进行任务之间的信息交互等。

## 致 谢

时间如流水一般奔流不停，转眼之间研究生学习生活即将结束，在这段时间里有过快乐，有过悲伤，有过甜蜜，有过苦涩，是一段让我终生难忘的经历，在这里要感谢那些给予过我帮助和关怀的老师、同学和朋友。

感谢我的导师李曦副教授，论文的研究工作至始至终都得到了他的悉心指导同时也凝聚着他的心血，李老师宽容、谦和的人格魅力让我倍感亲切，每当研究工作陷入困境的时候李老师都会给出一些有效的建议，使我常常有种醍醐灌顶的感觉，同时李老师对新事物和思想的敏锐的洞察力和探索精神也是我以后学习和工作上所要学习和努力的地方，在这里谨向李老师致以诚挚的谢意。

感谢李叶松教授和程善美教授，在论文的开题阶段他们给了我很多宝贵的意见，他们严谨的治学精神，精益求精的工作作风，深深地感染和激励着我，让我受益匪浅。

感谢师兄曹红亮和蒋建华，每当我遇到学习和生活中的困难时，他们都会给予我指导和帮助，谢谢这两年半时间他们对我的照顾；感谢同窗李想对我的鼓励和帮助；感谢师弟胡刚、蔡于勋、黄捷、秦强、汪宏斌和师妹张琳在很多事务上的分担，和你们在一起使我的业余生活更加丰富多彩；感谢杨云海、李文虎、张宁云等实验室其他同学，与他们在一起让我度过了难忘的求学阶段。

最后感谢我的父母，他们不仅在生活上给予我悉心的关怀，而且在精神上给了我极大的鼓励，是他们给了我前进的动力；在这里谨向他们表达深深的感激之情。

马腾

2011年1月

## 参考文献

- [1] 姚思童, 司秀丽, 杨军,等. 燃料电池的工作原理及其发展现状[J]. 沈阳工业大学学报, 1998, 20: 42-45.
- [2] 魏增福, 郑金. 燃料电池发电的研究现状与应用前景[J]. 广东电力, 2009, 22: 1-6.
- [3] 李箭. 固体氧化物燃料电池: 发展现状与关键技术[J]. 功能材料与器件学报, 2007, 13: 683-689.
- [4] 韩敏芳,尹会燕,唐秀玲等. 固体氧化物燃料电池发展及展望[J]. 真空电子技术, 2005, 28: 23-26.
- [5] 黎永祥, 姚丹丹. 新型化学电源-燃料电池[J].湖南工业职业技术学院学报, 2006, 6: 31-35.
- [6] 詹姆斯.拉米尼, 安德鲁.迪克斯. 燃料电池系统-原理、设计、应用[M]. 北京: 科学出版社, 2006: 43-48.
- [7] Larminie J, Dicks A. Fuel Cell Systems Explained. 2nd ed. Chichester, West Sussex PO19 8SQ, England: John Wiley & Sons Ltd; 2003.
- [8] [http://www.corporate-ir.net/ireye/ir\\_site.zhtml?ticker=fcel&script=410&layout=-6&item\\_id=989377](http://www.corporate-ir.net/ireye/ir_site.zhtml?ticker=fcel&script=410&layout=-6&item_id=989377)
- [9] <http://fuelcellsworks.com/news/2009/10/01/mhi-achieves-3000-hour-operation-unprecedented-in-japan-for-sofc-mgt-combined-cycle-power-generation-system/>
- [10] <http://www.bloomenergy.com/news/>
- [11] [http://www.gcl-power.com/news/search\\_detail.asp?id=121.2004](http://www.gcl-power.com/news/search_detail.asp?id=121.2004)
- [12] 王树东, 周谨. 氢能与燃料电池的现状与未来. 我国能源可持续发展战略的专题研究第三专题氢能 战略研究(之二), 2004, 1-104
- [13] 王礼进, 张会生, 翁史烈. 内重整固体氧化物燃料电池控制策略研究. 中国电机工程学报 2008;28(20): 94-98.
- [14] Zhang H, Wang L, Weng S, Su M. Performance research on the compact heat

- exchange reformer used for high temperature fuel cell systems. *Journal of Power Sources* 2008;183(1): 282-94.
- [15] Mueller F, Jabbari F, Gaynor R, Brouwer J. Novel solid oxide fuel cell system controller for rapid load following. *Journal of Power Sources*, 2007;172(1): 308-23.
- [16] Mueller F. The Dynamics and Control of Integrated Solid Oxide Fuel Cell Systems: Transient Load Following and Disturbance Rejection. PH.D., University of California, Irvine; 2008.
- [17] Zhang XW, Chan SH, Ho HK, Li J, Li G, Feng Z. Nonlinear model predictive control based on the moving horizon state estimation for the solid oxide fuel cell. *International Journal of Hydrogen Energy* 2008;33(9): 2355-66.
- [18] SEDGHISIGARCHI K, FELIACHI A. H-infinity controller for solid oxide fuel cells[C]//Proceedings of the 35th Southeastern Symposium on System Theory, USA: Country of publication USA, 2003, 16-18:464-467.
- [19] AGUIAR P, ADJIMAN C S, BRANDON N P. Anode-supported intermediate temperature direct internal reforming solid fuel cell II. Model-based dynamic performance and control[J]. *Journal of Power Sources*, 2005, 147(1-2): 136-147.
- [20] 方康玲主编. 过程控制系统[M]. 武汉: 武汉理工大学出版社, 2007, 9~10.
- [21] Zhong Z-D, Zhu X-J, Cao G-Y, Shi J-H. A hybrid multi-variable experimental model for a PEMFC[J]. *J Power Sources* 2007, 164: 746-51.
- [22] Takagi T, Sugeno M. Fuzzy identification of systems and its application to modeling and control [J]. *IEEE Transactions of systems, Man and Cybernetics*, 1985. vol.15(1), pp.116-132.
- [23] Fu Lai Chung, Tong Lee. Fuzzy Competitive Learning[J]. *Neural Networks*, 1994, 7(3): 539-551.
- [24] 侍寿永. 基于PLC的校园多功能控制系统设计[J]. *自动化应用*, 2010,11: 4-6.
- [25] 彭欣芸,胡泽,席怀强,张韬. 基于松下PLC的无线远程监控系统的设计[J]. *仪器仪表用户*, 2010,15:2-4.
- [26] 张威. Stateflow逻辑系统建模[M]. 西安: 西安电子科技大学出版社, 2007,4-8.
- [27] 文凌波,王玉海,李兴坤,薛春宇. 基于MATLAB/STATEFLOW的AMT控制策略仿

- 真系统[J]. 车辆与动力技术,2005,1:3-4.
- [28] Zuliani P, Platzter A, Clarke EM. Bayesian Statistical Model Checking with Application to Stateflow/Simulink Verification[J]. 13th International Conference on Hybrid Systems - Computation and Control Stockholm, SWEDEN, APR 12-15, 2010.
- [29] 邵惠鹤. 工业过程高级控制[M]. 上海: 上海交通大学出版社, 1997,32-33.
- [30] 刘喜梅,张茜,郭静. PID控制在反应釜温度控制中的应用[J]. 微型机与应用, 2010, 20:3-5.
- [31] 杨向忠, 安锦文, 崔文革. 快速控制原型仿真技术应用[J]. 航天控制, 2009,2: 1-3.
- [32] 欧健,康小鹏,张勇,邓国红. 汽车稳定性快速控制原型设计[J]. 计算机仿真, 2009,6:1-3.
- [33] 蔡光权,张书朋,张云龙. 快速控制原型ADX系统的应用研究[J]. 小型内燃机与摩托车, 2010, 4:3-6.
- [34] 杨涤, 李立涛等. 系统实时仿真开发环境与应用[M]. 北京: 清华大学出版社, 2002,50-55.
- [35] 李中健, 杜娟. XPC驱动程序开发及在无人机仿真中的应用[J]. 计算机策略与控制, 2009,17:1-2.
- [36] 陈永春. 从Matlab/Simulink模型到代码实现[M]. 北京: 清华大学出版社, 2002, 25-30.
- [37] 苗立东. xPC目标驱动程序开发中的关键问题研究[J]. 计算机工程, 2009, 19: 3-4.
- [38] 曾繁素, 冯保初. PCI总线与多媒体计算机[M]. 北京: 电子工业出版社, 1998, 12-16.
- [39] 吴佳楠, 王旭永, 张红伟. 采用CMEX S函数编写XPC环境下的设备驱动模块的研究[J]. 测控技术, 2006, 25: 55-57.
- [40] Jean J.Labrosse,邵贝贝等译. 嵌入式实时操作系统uC/OS-II[M]. 北京航空航天大学出版社, 2003,31-33.

- [41] 吴永明,罗海据.  $\mu\text{C}/\text{OS-II}$ 系统中任务调度与监控机制改进[J].计算机工程,2009, 2: 2-4.
- [42] 吴勇. 高速嵌入式系统研究与实现[D].北京工业大学, 2003, 33-34.
- [43] 陶保壮.基于ARM7的嵌入式操作系统 $\mu\text{C}/\text{OS-II}$ 平台的剖析与实现[J].安徽工程科技学报,2004,4:1-2.
- [44] 季虹,付少锋,车向泉,周利华. 实时嵌入式操作系统 $\mu\text{C}/\text{OS-II}$ 内核的分析与改进[J]. 计算机工程, 2007, 10: 5-6.
- [45] 佟国香,李琦,李广军.  $\mu\text{C}/\text{OS-II}$ 嵌入式操作系统的多任务应用程序开发[J]. 光电工程, 2004, 6: 1-2.
- [46] 郑映, 张祖平. 基于ARM+ $\mu\text{C}/\text{OS-II}$ 的程序动态加载实现方案[J]. 舰船电子工程, 2009, 5: 3-5

## 附录 攻读学位期间发表的论文

- [1] 马腾, 李曦, 祝得治, 张瑞民, 付常亮. 固体氧化物燃料电池电堆电特性的建模. 电源技术 (已录用)