

西北工业大学

博士学位论文

(学位研究生)

题目：容错控制方法及在飞行
控制系统设计中的研究

作者：穆旭

学科专业：控制科学与工程

指导教师：章卫国

西北工业大学自动化学院

2010年 6 月

Northwestern Polytechnical University

In Partial Fulfillment of the Requirement

for the Degree of

Doctor of Philosophy

Fault-Tolerant Control Methods and its research in Flight Control System Design

By

Mu Xu

Supervisor: Zhang Wei-guo

School of Automation

Northwestern Polytechnical University

June, 2010

摘要

现代飞行控制系统日趋庞大和复杂,由于其特定的工作环境和工作任务,这类系统一旦发生事故就有可能造成人员和财产的巨大损失,飞行控制系统的安全性、可靠性已成为系统设计中必须首先考虑的关键技术问题。容错控制可以在发生故障的情况下,自动补偿故障的影响,维护系统的稳定性,并且尽可能的恢复系统正常工作时的性能,从而保证系统的运行稳定、可靠。容错控制为提高复杂飞行控制系统的可靠性和安全性开辟了一条新的途径。

本文首先介绍了当前容错控制的主要方法及其设计思想,在此基础上深入研究了特征结构配置重构设计、简单自适应容错控制、基于神经网络辨识器的容错控制和神经网络自适应容错控制等方法在飞行控制系统容错控制设计中的应用,本文的主要研究内容和创新点如下:

1. 研究了飞行器的传感器故障、作动器故障和一般部件故障,并建立了相应的数学模型,不同的系统故障模型之间可以相互转化,这给故障检测和诊断的研究带来了很大的便利。在此基础上综述了几种主要的故障检测与诊断技术和当前飞控系统自适应容错控制的主要方法和理论。

2. 提出了一种基于特征结构配置的容错控制方法。该方法通过重新设计输出反馈矩阵,使得故障后系统的闭环特征值和特征向量最大程度地恢复到故障前的特征值和特征向量,从而保证系统的性能得到最大程度的恢复。通过对系统进行响应分析,本文提出一种综合了特征值灵敏度和特征向量的寻优设计指标,在此基础上结合改进的粒子群算法,提出一种特征结构配置重构控制设计方法,有效解决了故障可能引起的维数不匹配情况下的特征结构配置设计,以及传统的基于输出反馈的特征结构配置重构方法仅能配置 $\max(m, p)$ 个特征值而可能引起的系统不稳定性问题。

3. 基于指令跟踪发生器理论,研究了简单自适应容错控制方法。这种设计方案无需飞行系统具体的故障信息,流程简单,设计方便,方法固定,适用于飞行系统的实时容错控制。本文分别采用了前向补偿器和反馈补偿器保证了系统的严格正实性;在自适应控制器部分,引入误差积分抑制外部扰动,改进了简单自适应控制器的鲁棒性,并进一步针对 PI 型简单自适应控制律存在着对被控对象要求过严和控制律计算困难等不足,提出一种变步长修正的简单自适应容错控制方法,保证系统输出跟踪理想参考模型的输出。

4. 研究了神经网络辨识器在飞控系统故障诊断与自适应重构中的应用, 提出一种对神经网络的结构和权值并行训练的学习策略, 该策略采用爬山思想改进了粒子群算法, 提高了粒子群的效率和解的质量, 以此训练神经网络的权值; 结合蚁群算法的信息素思想调整神经网络隐层结构, 充分利用了群体的搜索经验, 实现了优化过程中不同结构间信息的共享, 从而完成神经网络结构和权值的并行优化。在此基础上, 离线建立了神经网络辨识器模型; 然后根据飞控系统实时、闭环的特点, 结合残差序贯概率比假设检验, 采用神经网络辨识器对飞行系统进行故障隔离与自适应重构设计,

5. 针对非线性不确定的飞行控制系统故障, 结合神经网络和自适应控制的思想, 提出一种小脑模型神经网络 (CMAC) 自适应鲁棒容错飞行控制方法。CMAC 是一种局部逼近神经网络, 可调参数少, 学习速度快, 相比其他的神经网络更适用于实时控制; 为了保证系统的输出跟踪和神经网络学习收敛, 该方法采用反步递推 (backsteeping) 设计了自适应容错控制律和神经网络权值调整算法, 将整个系统的设计问题分解为一系列低阶子系统的设计问题, 从高阶系统的内核开始 (通常是系统输出的要求), 为每个子系统设计部分 Lyapunov 函数和中间虚拟控制量, 一直后推到整个系统, 将他们集成起来完成整个控制律的设计。采用 backsteeping 设计的控制器收敛速度很快, 非常适用于实时的容错控制。该方法中 CMAC 同时学习故障后的不确定性和虚拟控制量的导数信息, 避免了 Backsteeping 的计算膨胀问题。这种容错控制策略为 Control Lyapunov 函数的构造提供了一类较为简便的结构化、系统化方法, 保证了系统的输出跟踪和神经网络学习收敛。仿真结果表明, 容错控制达到了理想的效果。

关键词: 飞行系统, 容错控制, 重构, 特征结构配置, 粒子群算法, 简单自适应, 神经网络, 反步递推设计方法

Abstract

The modern flight system is gradually huge and complex, in view of its specific working environment and missions, it has the possibility that faults of this kind of systems will lead to heavy personnel and property loss. So security and reliability becomes the key technology during flight control system design. Fault-tolerant control is the technology that design control laws to make the aircraft maintain the safe capability and acceptable dynamic performance when any fault occurs. Fault-tolerant control technology opens up a new way to enhance the complex flight control system's reliability and the security.

This dissertation firstly introduces the main control methods and the design concepts of Passive FTC and Active FTC; further discusses the fault tolerant flight control design schemes based on Eigenstructure Assignment method, Simple Adaptive Control, Neural Network control and Backstepping algorithm. The contexts and the innovations of this paper are listed as follows:

1. Considered the characteristics of modern flight control system, the dissertation analysis Sensor faults, Actor faults and the general Fault type. On this basis, the corresponding mathematical models of flight fault are given. The different fault models can be transformed into each other, which makes the fault detection and diagnosis research easier. Then the dissertation introduces several major FDD technologies and main methods and theories of adaptive fault-tolerant control for the flight control system.

2. The dissertation presents a fault tolerant control design technique using Eigenstructure assignment method for FCS. This method redesigns the output feedback matrix which makes the fault system's Eigenvalues and Eigenvectors close to the ones of normal flight system to the greatest extent. So the stability and dynamic performance of fault systems are well recovered in this way. Considering the possible unobservable or uncontrollable modes caused by flight faults, the paper designs an Eigenstructure Assignment method based on Singular Value decomposition algorithm for the fault tolerant control. Further more, this paper presents an optimization design index combining Eigenvalues' Sensitivity and Eigenvectors, then adopts an improved Particle Swarm Optimize Algorithm to propose a new Eigenstructure Assignment Reconfiguration method. This design scheme is capable of dealing with

order-change problems that may occurs in an after-fault system, and all the set of nominal closed-loop Eigenvalues can be recovered in the case of output Eigenstructure Assignment. Finally, simulation results are provided to show the effectiveness of the proposed methods for FCS.

3. Compared to the regular Model-Reference Adaptive Control method, the dissertation studies a fault tolerant control design technique using Simple Adaptive Control on basis of Command Generator Tracking theory. In this method no exact failure information is required for the reconfiguration; moreover the design process is simple and convenient. So Simple Adaptive method is appropriate to the real-time fault tolerant control. At first, a feedback compensator or a feedforward compensator are designed by utilizing optimization method in order to ensure the strictly positive real of the failure plants. In this case, Simple Adaptive Controller is able to ensure the stability and good tracking performance of fault system. To enhance the robustness of fault-tolerant controller, the error integral method is adopted to improve the adaptive controller. On the other side, compared to the former PI design pattern, a new variable step-size modified Simple Adaptive Fault-Tolerant Controller is proposed and adopted to adjust the adaptive parameters. This approach can track the reference model and decrease the control Complexity.

The scheme is illustrated through flight control simulations. The results show that the aircraft still has good dynamic performance in the presence of control surface failure.

4. On basis of observer theory, the dissertation proposes a robust adaptive fault isolation and reconfiguration control design technique with neural network. The concept of this scheme is that neural network can be trained equal to normal system. Considering the real-time and closed loop feedback in flight control system, we build neural network observer offline and use it online for fault detection and fault isolation according to residuals sequential probability ratio test. If judging any fault occurs, neural network is adopted in the closed loop replacing the fault section. Moreover, for the consideration of slow convergence speed and easily getting into local minimum of traditional neural network weight learning algorithm, a new training algorithm for neural network based on improved particle swarm optimization and Pheromone Theory of Ant Colony Algorithm is presented. Finally, we use this scheme for sensor fault isolation and reconfiguration controller design. The simulations results show the validity of this method.

5. Considering the nonlinearity and uncertainty of Flight after-fault systems, the dissertation presents an adaptive fault tolerant flight control design technique using Cerebellar Model Articulation Controller. Cerebellar Model Articulation Controller is a kind of Local approximation neural network, so it's appropriate to real-time control. To ensure the system is stable and the training process of CMAC is converged, the paper uses the Backstepping algorithm to design the adaptive fault tolerant control law and the Training Algorithm of CMAC. Backstepping algorithm decomposes the whole system into a series of subsystems, and beginning with the output requirements, designs Lyapunov Function and virtual control input for each subsystem which integrates the whole system's control law. In this method, CMAC online approximates the after-fault system uncertainties and the derivative of virtual control input to avoid the calculation expansion problem of traditional Backstepping algorithm, which is due to derivate the virtual control input. This fault tolerant control algorithm provides a structured and systematic approach for setting up Control Lyapunov Function, and ensures the after-fault system output tracing and CMAC Convergence. The simulation shows that fault tolerant control achieves the ideal effect.

Keywords: flight system, fault-tolerant control, reconfiguration, Eigenstructure Assignment, Particle Swarm Optimize Algorithm, Simple Adaptive control, Neural Network, Backsteeping design

目 录

摘 要.....	I
Abstract.....	III
目 录.....	VII
第一章 绪 论	1
1.1 研究背景及意义	1
1.2 容错控制.....	2
1.2.1 被动容错控制 (passive FTC)	3
1.2.2 主动容错控制 (active FTC)	4
1.2.3 容错控制研究的若干问题讨论	14
1.3 本文的内容安排	16
第二章 飞行控制系统故障诊断与容错控制的基本概念	19
2.1 引言	19
2.2 飞机常见故障建模	20
2.3 故障检测与诊断技术	25
2.4 飞行控制系统的容错控制方法	27
2.5 小结	30
第三章 基于特征结构配置的容错飞行控制设计.....	31
3.1 引言	31
3.2 基于特征结构配置的重构控制	33
3.2.1 问题描述	33
3.2.2 特征结构配置重构算法实现	35
3.3 基于改进粒子群优化的特征结构配置重构控制律设计.....	40
3.4 仿真算例.....	45
3.5 小结	50
第四章 基于简单自适应的容错飞行控制设计.....	51
4.1 引言	51

4.2 简单自适应容错控制	53
4.2.1 问题描述	53
4.2.2 基于指令跟踪发生器理论的简单自适应控制	53
4.2.3 简单自适应鲁棒容错控制设计	57
4.2.4 基于变步长修正的简单自适应容错控制设计	58
4.3 简单自适应方法的正实性条件	60
4.3.1 应用前馈补偿器保证正实性条件	61
4.3.2 应用反馈补偿器保证正实性条件	62
4.4 简单自适应容错飞行控制系统设计	63
4.5 仿真算例	64
4.6 小结	68
第五章 基于神经网络辨识器的容错飞行控制设计	69
5.1 引言	69
5.2 神经网络学习算法研究	70
5.2.1 BP 神经网络的基本结构和学习算法	70
5.2.2 基于蚁群和改进粒子群的神经网络学习算法	72
5.3 基于神经网络辨识器的容错控制设计	77
5.3.1 神经网络辨识原理	77
5.3.2 基于神经网络辨识器的自适应容错控制	79
5.4 仿真算例	81
5.5 小结	86
第六章 基于神经网络控制器的容错飞行控制设计	87
6.1 引言	87
6.2 小脑模型神经网络	89
6.3 模型参考自适应逆神经网络容错控制系统	93
6.3.1 模型参考自适应逆控制	93
6.3.2 基于模型参考自适应逆的神经网络容错控制设计	94
6.4 反步递推自适应算法	95
6.4.1 问题描述	96

6.4.2 基于 backstepping 算法的神经网络容错飞行控制方法	99
6.5 仿真算例	105
6.6 小结	109
第七章 总结与展望	111
7.1 工作总结	111
7.2 工作展望	112
参考文献	115
博士在读期间发表论文和参加科研情况说明	129
致 谢	131

第一章 绪 论

1.1 研究背景及意义

近年来随着科学技术的迅速发展,系统的自动化程度越来越高,规模也日趋庞大和复杂,人们对系统可靠性提出了较高的要求。特别是现代飞行控制系统,由于其特定的工作环境和工作任务,这类系统一旦发生事故就有可能造成人员和财产的巨大损失,飞行控制系统的安全性、可靠性已成为系统设计中必须首先考虑的关键技术问题。1997年9月,印度尼西亚“鹰记”航空公司的一架A300在印尼以北苏门答腊首府棉兰附近坠毁,这是印尼1990年以来最大的一起空难,飞机失事与棉兰上空烟雾弥漫、能见度低等因素有关,大型客机在浓雾气象条件下着陆,能见度几乎为零,飞行员只能依靠自动着陆系统着陆,此时控制系统一旦发生故障,就会导致机毁人亡;1999年,埃及航空公司的波音767-330型客机在起飞后不久便坠落在美国马萨诸塞州以东约100公里的大西洋上,根据黑匣子的分析结果,飞机控制技术故障最有可能是坠毁的原因……这些惨痛教训使人们更进一步认识到在大型复杂飞行系统中提高控制系统可靠性与安全性的重大意义。

在这样一种背景下,容错控制FTC(Fault Tolerant Control)技术的研究被推到了前沿位置,并得到了飞速的发展。“容错”是容忍故障的简称,容错控制是指控制系统在发生故障的情况下,能够自动补偿故障的影响,维护系统的稳定性,并且尽可能的恢复系统正常工作时的性能,从而保证系统的运行稳定、可靠。容错控制为提高复杂系统的可靠性和安全性开辟了一条新的途径。

容错控制的思想最早可以追溯到1971年,Niederlinski首先提出完整性控制(Integral Control)概念^[1],Beard也同时开始了容错控制的研究^[2]。1986年,美国国家科学基金会和IEEE控制系统学会在美国Santa Clara大学联合召开了关于控制所面临的机遇与挑战的讨论会,来自世界各地的52位控制理论与应用领域的权威到会,会议正式提出容错控制的概念,并把多变量鲁棒、自适应和容错控制列为三大富有挑战性的研究课题。

容错控制迄今只有30多年的历史,却取得了一系列理论上和工程上的成果。促使容错控制迅速发展的一个最重要的动力来自于航空航天领域,欧美国家从20世纪70年代起就不断投入巨资对容错控制技术进行研究,为研制具有强容错

能力的高性能飞行器服务，期望达到在飞行系统某个部件故障，甚至多个部件故障时，依靠容错控制技术也能够保证飞行器的生存能力和较理想的飞行特性。作为一门新兴交叉学科，容错控制与鲁棒控制、故障检测与诊断、自适应控制、智能控制等有着密切的联系。它的理论基础涉及多门学科，如现代控制理论、信号处理、模式识别、人工智能、模糊集理论、最优化方法、决策论、统计数学等。近年来，容错控制已得到了国际自动控制界的高度重视，成为自动控制的一个重要分支。

1.2 容错控制

容错控制是在系统某些部件发生故障的情况下，保证系统的稳定性，并使得故障系统仍具有较理想的特性。容错控制设计有硬件冗余和解析冗余两种方式。

最早的容错技术都是硬件冗余，这种策略源于可靠性分析理论，即并联系统可以大大提高系统的可靠性，对于重要部件采用多重备份的办法来提高系统的容错性能。硬件冗余大大提高了系统的可靠性，在工程实际中具有广泛的应用。然而硬件冗余容错控制需要耗费较多的成本，占用更大的硬件空间，对复杂庞大的系统采用过多的硬件冗余是不现实的，因此随着基于模型的故障诊断方法的发展，基于解析模型的容错控制技术也逐步发展起来。

解析冗余容错控制通过设计容错控制器提高控制系统的冗余度来改善系统的容错性能。按照控制器设计方法的不同，容错控制可以分为被动容错控制和主动容错控制。

被动容错控制是设计适当固定结构的控制器，控制器除了考虑正常工作状态下的参数值，同时还要考虑可能的故障特点和系统不确定性的影响，利用鲁棒控制技术使得闭环系统对某些故障不敏感。被动容错控制在正常工作状态和故障状态使用同样的控制策略，不需要在线调整控制器结构或参数，当预期的故障如执行器、传感器和其他部件失效时，仍能保证系统的稳定性和较理想的性能，因此被动容错控制又称为鲁棒容错控制。

主动容错控制是在故障发生通过重新调整控制器的参数，或者改变控制器的结构，保证故障后系统的稳定性和可靠性，因此主动容错控制又称为自适应容错控制。故障诊断（Fault detection and diagnosis）和重构控制（Reconfigurable Control）是主动容错控制的主要内容。故障诊断技术是在现代控制理论、数理统计、人工智能等方法分析、处理非正常状态下系统特性的基础上发展起来的；重构控制则保证当系统中的某些部件发生故障时，系统仍能按期望的性能指标完成任务。有的主动容错控制方法需要故障诊断子系统，有的主动容错控制方法不需

要故障诊断子系统，但需要已知各种故障的先验知识，目前主动容错控制在工程实际中具有广泛的应用。

1.2.1 被动容错控制 (passive FTC)

被动容错 (passive fault tolerant control) 的核心思想是使闭环系统对不确定性和一些可能故障具有鲁棒性。目前，被动容错控制方法主要有完整性控制、可靠镇定和同时镇定。

完整性控制 (integrity control) 的概念是由 Niederlinski 在 1971 年提出的^[1]，这是容错控制领域中研究最早方法，也一直是被动容错控制研究中的热点问题^[12~16]。完整性指的是多变量线性定常系统在传感器和执行器故障的情况下，利用正常工作的传感器和执行器代替故障元件的功能，闭环系统仍然是稳定的。据研究表明系统故障的 60% 以上是发生在控制系统，而控制系统的故障多为传感器故障和执行器故障，因此，完整性控制具有很高的应用价值和工程意义。

目前国内外关于完整性控制问题的研究取得了较多的研究成果，然而完整性控制问题的讨论还远未彻底解决，主要问题是对高维多变量系统缺乏有效求解完整性控制律的构造性方法。

可靠镇定是对控制器失效的容错控制，最早由 Siljak.D.D 在 1980 年提出^[3]，这种技术采用多个补偿器并行镇定同一个被控对象，当其中某些补偿器失效时，其余的补偿器正常工作，从而保证闭环系统仍然是稳定的。经过学者们大量的深入研究，至今可靠镇定技术已基本趋于成熟^[4,6,7]。

同时镇定，又称为联立镇定、多模型设计方法，是对被控对象故障的容错控制，最早由 Ackemann、Saeks、Vidyasagar^[4,5]等人提出，指的是针对多个被控对象，构造一个固定的控制器，可以镇定其中的任一被控对象。同时镇定具有两个主要作用^[20,80]：一是当被控对象发生故障时，可以使其保持稳定，具有容错控制的作用；二是对于非线性被控对象，常用的设计方法是将其线性化在某个工作点上，然后采用线性控制方法进行控制，当工作点变动时，其线性模型也会发生改变，采用同时镇定可以有效的镇定非线性对象的不同线性化模型，保证对非线性对象的控制效果。

同时镇定的设计目标是建立一个公共的状态反馈控制器，使之可以稳定大多数故障情况下的系统模型，并且保证故障系统具有较为理想的动态、静态品质特性。由于同时镇定工程意义明显，成为容错控制的一个重要研究方向。文[8]是最早开始研究同时镇定问题的文章，随后学者们出现了许多更深入的研究，目前同时镇定容错控制大多数是基于互质分解方法或者基于稳定性理论设计^[9~11]。

1.2.2 主动容错控制 (active FTC)

主动容错控制 (active fault tolerant control) 是故障情况下的强自适应控制, 通过系统故障检测与诊断 (FDD, Fault Detection and Diagnosis), 当检测到系统发生故障后, 根据不同的故障信息, 设计相应的容错控制律, 重新调整控制器的参数, 必要时还会改变控制器的结构, 以保证系统的稳定性和理想的性能指标。

相比而言, 虽然被动容错控制在故障发生后可以及时实现容错控制, 不存在主动容错控制中因为故障隔离延而引起的控制性能变坏问题, 但是被动容错控制设计方法只能适应事先考虑到的故障情况, 对于预想故障种类较多或者未知故障发生时, 被动容错控制的设计过程通常会很复杂或者缺乏容错能力; 并且被动容错控制要求即使在系统正常工作的情况下, 控制器也要兼顾故障条件下的控制要求, 而系统大部分时间都是工作在正常的状态下, 这显然是对系统的一种过高要求, 其容错性往往以牺牲系统性能为代价。相比之下, 主动容错控制更具有生命力。多数主动容错控制需要故障诊断子系统, 少部分不需要故障诊断子系统, 但需要已知各种故障的先验知识。故障检测与诊断、系统重构是主动容错控制的两大课题。

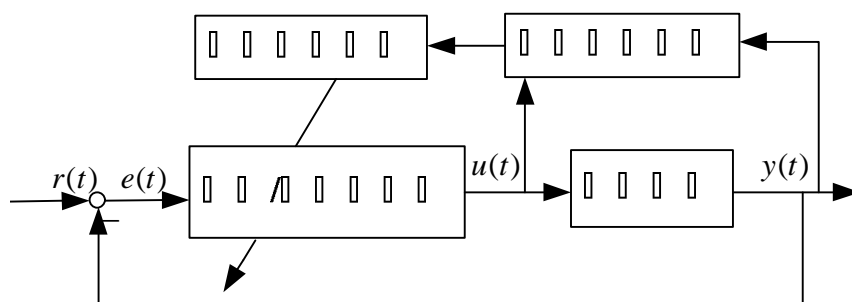


图 1-1 主动容错控制

1.2.2.1 故障检测与诊断技术

故障检测与诊断是在一定的检测策略的指导下, 自动检测系统, 获取诊断对象的故障模式, 提取故障特征, 在此基础上根据推理判断原则对故障信息做出综合评估, 并向控制器提示所要采取的措施。控制系统故障诊断技术是一门应用型的边缘学科, 它的基础理论是现代控制论、计算机工程、数理统计、信号处理、模式识别、人工智能、人工神经网络以及相应的应用学科。目前故障诊断的研究已经远远超前于容错控制的研究, 成为控制领域中的前沿课题。

故障检测与诊断技术的发展大体经历了几个阶段: 最初故障检测与诊断主要依靠人的感觉、经验及简单仪表, 对诊断信息也只做简单的数据处理; 70 年代, 随着传感器技术、动态测试技术和信号分析技术的发展, 解析冗余代替硬件冗余

^[2]，通过比较器的输出得到系统故障信息标志着故障检测与诊断技术的开端，在这一期间学者们提出一些简单的故障诊断算法，如检测滤波器、广义似然比、极大似然比；20 世纪 80 年代以来，故障检测与诊断理论蓬勃发展，逐渐形成相对清晰的设计思路，特别是基于观测器/滤波器方法、系统辨识和参数估计方法在工程中得到广泛的运用；近年来，控制系统故障诊断理论的研究更加深入，各种方法相互渗透和融合，应用的领域也有了很大的扩展。参数估计和观测器/滤波器方法仍然是使用得最多的故障诊断方法，随着系统日趋庞大复杂化，为适应现代大型系统故障检测与诊断的需求，基于神经网络、模糊逻辑、专家系统及组合的故障诊断方法迅速发展，并且在维修工程及可靠性工程中得到了广泛应用。

故障建模、故障检测、故障的分离与估计和故障的评价与决策是故障诊断技术的主要内容。故障建模是根据先验信息和系统输入/输出关系，建立系统故障的数学模型，作为故障检测与诊断的依据；故障检测是根据估计变量判断运行的系统是否发生故障，一旦判断系统发生故障，就会发出报警；故障的分离与估计是指如果系统发生了故障，定位故障源，分析故障性质是执行器、传感器、被控对象还是特大扰动等，同时计算故障的程度、大小及故障发生的时间等参数；故障的评价与决策根据故障分离与估计的结论对故障的严重程度、故障对系统的影响和发展趋势作出评价，从而决定是否需要容错控制。

故障检测与诊断技术发展过程中，在不同的应用领域，与不同的学科相结合产生了多种故障诊断方法。目前，故障诊断方法可以分成三大类：基于模型的故障诊断方法、基于信号处理的故障诊断方法和基于知识的故障诊断方法。

（一）基于模型的故障诊断

基于模型的故障诊断方法起源于 1971 年的 Beard 发表的博士论文中^[18]，是故障诊断技术中开始最早、也是目前研究最深入、成果最多的方法。基于模型的故障诊断技术通过构造观测器估计出系统输出，与系统输出的实际测量值做比较得到残差信号。残差信号中包含着丰富的故障信息，通过对残差信号的持续观测，可以对故障进行判决和定位，达到故障检测和诊断的目的。基于模型的故障诊断方法充分利用系统内部的深层知识，更有利于系统的故障诊断，在故障诊断领域具有重要的地位。目前基于模型的故障诊断方法主要有：

（1）状态估计方法

1971 年，Bear 在他的论文中提出了故障诊断检测滤波器的概念^[18]，这标志着基于状态估计的故障诊断方法的诞生，随后学者们对这种方法进行了大量的研究。

状态估计方法是当系统状态可观和部分可观时，构造状态观测器对系统的状态进行估计，将估计值与实际测量值进行比较得到残差序列，对系统进行故障诊

断。在理想情形下，系统建模精确、不受外界扰动及测量噪声影响，无故障发生情形下残差信号为零，而当系统出现故障时，残差会显著偏离零，只需对残差信号进行故障方向辨识就可准确地定位出故障的位置。

在实际工程中，系统的精确数学模型一般很难得到，系统建模误差、不确定性扰动和测量噪声总是不可避免，因此即使在无故障发生的情形下，残差信号也可能是非零的^[19]。基于状态估计的故障诊断方法对故障信号检测的敏感性和其对系统参数摄动的鲁棒性是始终存在的一对矛盾，协调二者之间的关系以便能达到各自指标的最优化是状态估计方法关心的一个问题。目前该方法的研究主要集中在提高检测系统对建模误差、扰动和噪声等未知输入的鲁棒性以及系统对于早期故障的灵敏度问题。基于状态估计的故障诊断方法主要有自适应观测器方法、Kalman 滤波器方法、Luenberger 观测器方法、滑模观测器方法、模糊观测器方法等。

(2) 参数估计方法

参数估计方法^[171]的基本思想是：系统元部件的特性可以通过物理参数来描述，系统抽象出来的数学模型中的参数称为模型参数，模型参数与物理参数以一定的映射关系联系在一起。系统的故障是由于系统元部件的损坏、失效等性能改变引起的，系统发生故障时，必定导致系统某些元部件的物理参数发生了变化。若已知模型参数的变化情况，就能通过模型参数和物理参数的关联关系确定出物理参数的变化情况，实现根据参数的估计值与正常值的偏差情况判断系统故障情况的目的。由于非线性系统的数学模型较难建立，相比状态估计方法，参数估计方法更适用于非线性系统的故障诊断问题。基于参数估计的故障诊断主要有最小二乘法 LS、递推最小二乘法 LS、强跟踪滤波器方法。其中强跟踪滤波器方法^[172]是参数估计中的一个重要方法，该方法采用了强跟踪滤波器来估计模型参数，对不确定模型具有鲁棒性，同时对突变和缓变状态具有很好的跟踪能力，适用于传感器故障、执行器故障和其他元器件故障。

状态估计和参数估计各有优缺点^[20]。基于状态估计的故障诊断收敛快，具有较好的实时性，而基于参数估计的故障诊断方法收敛较慢，容易造成故障诊断的延时；基于参数估计的方法需要有激励信号存在，状态估计方法对系统输入信号的要求不是很严格，不需要有连续不间断的激励信号存在；状态估计方法能够检测出来的故障是有限的，特别是被控对象的故障情况，即便是能检测出来，也很难准确定位出被控对象中受损元件的物理位置以及受损程度，而参数估计方法很容易定位故障情况，同时参数估计方法对模型不精确具有鲁棒性。这两种故障检测方法可以互补对方的劣势，将二者结合起来使用可以更好的进行故障诊断，这是今后故障诊断方法的一个研究热点。

(3) 等价空间方法

等价空间方法又称为等价方程方法、奇偶空间诊断方法,是针对线性系统提出的一种故障诊断技术,该方法通过系统输入/输出的实际值,检验被诊断对象数学关系是否与正常工作状态等价,从而达到故障检测和分离的目的^[173]。

现阶段基于模型的故障诊断方法的研究成果仍然主要集中于线性系统,深入研究非线性系统的通用故障诊断技术具有重要的意义。

(二) 基于信号处理的故障诊断

随着系统日趋复杂,获得系统的数学模型在很多情况下是很困难的问题,因此不依赖系统模型的故障诊断方法引起了学者们的高度重视。基于信号处理的故障诊断方法不需要精确的数学模型,通过借用信号处理手段,对系统的输出在频率、幅值、相位和相关性等方面进行分析从而获取系统中的故障信息,对系统进行故障检测和故障分离。这类方法不仅适用于线性系统,而且还适用于非线性系统的故障检测和诊断,目前基于信号处理的故障诊断方法主要有直接测量系统输入输出的方法、基于小波变换的方法、主元分析法、信息匹配诊断法、基于信息融合的方法以及信息校核的方法等。

(1) 基于直接可测信号的故障诊断方法

基于直接可测信号的故障诊断方法又称为趋势检测法,该方法根据直接可测的输入、输出信号或者它们的变化趋势来进行故障诊断。在正常情况下,系统输入、输出信号的变化范围或者变化趋势总是在一定的范围之内,当超出规定的范围时,就说明系统中有故障发生^[174]。

(2) 基于小波变换的故障诊断方法

基于小波变换的方法^[22]是一种重要的基于信号处理技术的故障诊断方法。小波变换是一种时间-尺度分析方法,具有多分辨率分析的特点。利用连续小波变换可以检测信号的奇异性,区分信号突变和噪声;而离散小波变换可以检测随机信号频率结构的突变。基于小波变换故障诊断方法的基本思路是:首先对系统的输入输出信号进行小波变换,利用该变换求出输入输出信号的奇异点,然后去除由于输入突变引起的极值点,则其余的极值点对应于系统的故障。基于小波变换的故障诊断方法不需要系统的数学模型,对噪声的抑制能力强,有较高的灵敏度,运算量也不大,可以进行在线实时故障检测,是一种很有前途的方法。

(3) 主元分析法

主元分析法是利用过程的多余自由度来检测过程的异常,该方法的基本思想是:利用过程的历史数据建立正常情况下的过程主元模型,一旦过程的测量数据与主元模型发生冲突,就可以通过对测量数据的分析以分离出不同的故障。主元分析法主要用于数据中含有大量相关冗余信息时的故障检测与隔离。

基于信号处理的故障诊断方法实现简单，在过程监控中有着广泛的应用。但这种方法也有缺陷，该方法只有当故障发生到相当的程度并且影响到外部特征时才有效，而且基于信号处理的故障诊断技术只能对故障范围做出粗略的判断，大多数情况下并不能直接定位故障，同时其故障诊断的效果高度依赖数据的数量与质量。

（三）基于知识的智能故障诊断

20 世纪 80 年代，随着人工智能技术的迅速发展，人们对智能故障诊断技术进行了更加深入与系统的研究。一方面，智能故障诊断不依赖系统的数学模型，具有传统诊断方法无可比拟的优点，另一方面，复杂设备的故障诊断在很大程度上需要依赖专家的经验知识，因此基于建模处理和信号处理的诊断技术正发展为基于知识处理的智能诊断技术。基于知识的故障诊断方法与基于信号的故障诊断方法类似，也不需要精确的数学模型。二者的差别在于，基于知识的故障诊断技术充分挖掘了被诊断对象的深层知识，分析系统历史运行数据，利用故障信息，将系统对象的知识与专家诊断知识有效结合，大大提高了诊断的决策水平。

目前基于知识的故障诊断方法主要有专家系统故障诊断方法、基于神经网络的故障诊断方法、基于模糊数学的故障诊断方法、基于模式识别的故障诊断方法等。

（1）基于神经网络（neural network）的故障诊断

神经网络代表了一种新的方法体系，具有自学习和自适应性，该方法采用分布式信息存储，利用网络的拓扑结构和权值分布实现非线性映射，并利用全局并行处理实现从输入空间到输出空间的非线性信息交换，在处理非线性问题、难于建模的系统和故障繁多的复杂问题方面很有优势，非常适用于故障诊断。对于特定问题建立适当的神经网络故障诊断系统，可以从其输入数据（代表故障症状）直接推出输出数据（代表故障原因），从而实现离线（有导师的学习）或在线（无导师的学习）故障检测与诊断^[20]。神经网络在故障诊断中具体应用方式有：从模式识别角度，应用神经网络作为分类器进行故障诊断；从预测角度，应用神经网络作为动态预测模型进行故障预测；利用神经网络极强的非线性动态跟踪能力，进行基于结构映射的故障诊断；从知识处理角度，建立基于神经网络的诊断专家系统等。

基于神经网络进行故障诊断存在着难以利用经验知识、训练时间长、复杂系统稳定性分析困难、算法的收敛性和实时性难以保证等缺点，因此近年来越来越多的研究集中在将小波理论、模糊算法、进化算法、混沌理论等与神经网络结合使用的诊断技术。

（2）基于专家系统的故障诊断

专家系统是计算机技术和人工智能理论发展的结果,基于专家系统的故障方法充分利用了故障诊断实践经验和大量的历史故障信息,采用计算机技术建立故障检测诊断专家系统的数据库,其中存储了系统的故障征兆、故障模式、故障成因、处理意见等内容,在此基础上采用计算机程序对系统进行故障推理判断,智能地解决系统的故障诊断问题。基于专家系统的故障诊断方法是诊断领域引人注目的发展方向之一,也是研究最多、应用最广的一类智能诊断系统,主要用于很难建立数学模型的复杂系统。

(3) 基于模糊推理的故障诊断

基于模糊推理的故障诊断方法模拟了人类思维方式,应用模糊逻辑对故障输出的残差进行处理。模糊逻辑的引入主要是为了克服由于过程本身的不确定性以及噪声等所带来的困难,因而在处理复杂系统的大时滞、时变及非线性方面,显示出它的优越性,这种方法对于和模糊规则接近的故障模式诊断准确,非常适于含有模糊现象和不确定信息的系统的诊断,是目前故障诊断方法研究的一个热点。目前,基于模糊推理的故障诊断主要应用于残差的模糊逻辑评价和自适应调节残差的阈值^[175]。单纯采用模糊推理的故障诊断不具备自学习能力,因此无法对新故障做出诊断,同时这种方法的模糊规则、隶属函数和决策算法的最优化过多依赖于经验,因此近年来人们更多的是将模糊推理和神经网络、小波理论等算法结合,构成更完善的故障诊断技术。

故障诊断领域的权威——德国的 P.M. Frank 教授认为所有的基于解析冗余的故障诊断方法可以划分为基于解析模型的方法、基于信号处理的方法和基于知识的方法这三类^[21],然而故障诊断技术发展迅速,近年来又出现了新的概念和技术方法,比如基于观测器的复合方法、基于离散事件的诊断方法等,新方法对 P.M. Frank 教授的分类方法进行了扩充,文[23]在此基础上对故障诊断方法进行了更为细致的分类。

1.2.2.2 重构控制方法

主动容错控制是在故障信息的基础上,对系统进行重构控制,或调整控制器参数,或改变控制器结构,以使得故障后的系统保持稳定性,并满足一定的性能要求。

重构控制经过近年来的发展,取得了很多理论和工程上的成果,按照不同的重构规则,主动容错控制设计策略主要有控制律重新调度、控制律重构设计、模型跟随重组控制^[17,22,80]。

控制律重新调度是一种较为简单的主动容错控制方法,该方法的基本思想是离线计算出各种故障下所需的控制律增益参数,并将这些参数列表存储在控制计算中,系统根据在线的故障检测与诊断技术得到当前的故障信息,从而选择匹

配的增益参数,得到对各种故障的容错控制律。可见,采用实时的专家系统进行增益调度将会产生很好的效果^[22]。控制律重新调度的增益参数是离线设计好的,容错控制律制只对事先考虑到的故障具有容错能力。控制律重新调度可以看作是反馈控制系统增益或结构可由前馈补偿器调整,其增益的选择由故障诊断机构的输出控制。控制律重新调度方法很大程度上受故障诊断机构鲁棒性的影响,故障诊断的任何漏报和误报都可能对控制效果产生严重后果。

控制律重构设计是在系统故障后,根据故障的状况,在线重构控制律,以使故障系统尽可能的恢复到正常工作状态,或者在保证系统稳定的情况下,具有可接受的性能。这类重构方法是建立在 FDD 结果的基础上,目前控制律重构设计中研究较多的是控制器重构^[17,82]。

模型跟随重组控制是基于模型跟随自适应的思想设计控制器,无需故障诊断单元和具体的故障信息,保证被控对象的输出跟随理想参考模型的输出,这是目前很有生命力的一个研究方向,人们将模型跟随重组控制的设计思想和其他具体方法结合,产生了很多行之有效的重构设计方案,如多模型自适应控制,神经网络自适应控制等。

近年来,随着系统的日益复杂化和庞大化,系统精确的数学模型有时很难建立;并且很多时候系统表现出非线性,在这种背景下,非线性和不确定系统的容错控制成为人们关注的焦点。传统的容错控制方法很难有效解决这种问题,而智能控制如神经网络、专家系统、模糊逻辑等以其高度的自适应、自学习能力,成为容错控制的有利工具。

具体而言,目前的重构控制方法主要有:

(1) 伪逆法

在线性系统容错控制中,通过设计一个新的反馈增益矩阵实现重构是一个很重要的内容,目前主要有三种方法:线性二次最优调节器法 LQR、广义逆法、特征结构配置法^[17]。

伪逆法是广义逆法中常用的一种特例,是一种基于线性系统的重构控制方法,当操纵面发生故障时,可利用剩余无故障的操纵面进行适当的线性组合,重构故障操纵面信号,这通常表现为在原输入阵前乘一个伪逆阵,所以称其为伪逆法。伪逆法的应用是在故障发生时,首先进行故障的检测、隔离和估计,然后通过修改反馈增益,使得重构后的系统性能接近正常飞机性能。

伪逆法可对故障进行在线计算和调整,结构简单,应用方便,在早期得到了较多的采用。但伪逆法无法保证重构系统的稳定性,并且重构控制依赖于精确的故障检测和诊断系统,这些缺陷影响了该方法在工程上的应用。

(2) 线性二次型调节器 LQR

线性二次型调节器控制法通过求解故障系统的 Riccati 方程, 设计最优控制矩阵, 基于 LQR 的容错控制一般可以保证故障后闭环系统的稳定性; 但是由于 LQR 优化过程中权矩阵的选取概念模糊, 具有很大的不确定性和任意性, 因此 LQR 难以使故障系统恢复到正常工作状态下的性能。

(3) 特征结构配置

特征值配置法^[38,153-161]是针对于线性模型的重构控制方法。由线性系统的动态响应分析可知, 特征值决定了系统的稳定性和响应速度, 而特征向量则决定了各运动模态对系统响应的耦合程度。对系统进行特征结构配置, 可以更好的把握系统的性能; 对故障系统进行特征结构配置, 设计合适的反馈矩阵, 使其特征值和特征向量得到最大程度的恢复, 从而使得故障系统的性能接近正常工作状态, 这是特征结构配置重构方法的基本思想。

从工程实践角度来讲, 实际系统中并非所有状态可测, 或者状态可测但测量成本较高, 因此一般采用输出反馈进行特征结构配置, 但是这种方法并不能配置全部的闭环特征值而可能引起重构系统的不稳定性; 并且故障后可能存在系统的维数不匹配问题而导致算法失效, 针对这些问题, 本文在第三章进一步研究了基于输出反馈的特征结构配置重构方法, 通过对系统进行响应分析, 提出一种综合了特征值灵敏度和特征向量的设计指标, 在此基础上结合寻优算法, 提出一种有效的特征结构配置重构控制设计方法。

(4) 多模型自适应控制

多模型自适应控制方法是离线设计与在线控制的巧妙结合, 针对事先考虑到的故障, 分别离线设计重构控制律, 在飞行中根据故障的模型切换到相应的控制律, 从而完成对故障系统的容错控制。

多模型自适应控制^[165]的关键问题是故障模型的选取、相应的控制律和切换策略, 该方法利用多模型覆盖被控对象的不确定性, 把复杂系统转化为多个简单系统的组合, 将多个简单控制器有机结合成一种在大范围内具有高鲁棒性的控制系统, 不需要在线实时设计, 快速性好。尤其在故障少且故障信息比较充分的情况下, 多模型自适应方法能够快速、稳定地实现重构控制。但是当故障数目较多时, 所需要的工作量呈指数增加, 控制器的规模将变得十分庞大, 并且这种方法对未知故障缺乏容错能力。

(5) 模型参考自适应/简单自适应

模型参考自适应通过设置理想的参考模型, 实时调整控制律, 使系统的实际响应跟踪理想模型响应。对于隐模型跟踪, 需要快速的系统参数辨识; 对于直接的模型跟踪, 需要有一个理想结构; 自适应控制要求系统参数变化较慢, 以便有足够的时间进行预估^[103]。这种重构控制思想具有较好的鲁棒性, 是人们研究的

一个热点方向。本文在模型参考自适应的基础上,进一步研究了简单自适应容错控制。

简单自适应控制于 1979 年由 H.Kaufman, K.Sobel 等学者提出^[30],相比常规模型参考自适应解决问题的方法和过程,简单自适应方法不仅具有控制结构简单,可调参数少;控制系统的设计几乎与被控对象无关,控制器的设计不依赖于被控对象,只是利用参考模型和跟踪误差信息完成设计,实现系统对参考模型的渐进跟踪等优点,更具有适用于单变量系统和多变量系统;可实现高阶被控对象对低阶理想参考模型的性能跟踪;可以构成通用的自适应控制器应用于不同的被控对象等诸多优点。本文在第四章基于指令跟踪发生器理论,研究了简单自适应容错控制方法^[36,37]。这种容错设计方案无需飞行系统具体的故障信息,流程简单,设计方便,方法固定,适用于飞行系统的实时容错控制。

(6) 基于神经网络的重构控制

神经网络的研究始于 1943 年,至今已有 60 多年的历史。心理学家 McCulloch 和数学家 Pitts 提出第一个神经模型,该模型被称为 McCulloch-Pitts (MP) 神经元模型^[166],从此开创了神经网络理论研究的时代。1949 年,心理学家 Donal O. Hebb 提出了修正神经元连接强度的“突触修正假设”,并据此提出了神经网络的学习准则,这就是著名的 Hebb 学习准则^[167],Hebb 的研究成果为神经网络的学习算法奠定了基础,促进了神经网络研究的早期热潮。1982 年,美国加州工学院生物物理学家 Hopfield 提出 Hopfield 网络模型^[168],引入网络能量函数概念,并对该模型给出了稳定性判据,这标志着神经网络研究第二次高潮的到来,从此揭开了神经网络计算机研究的序幕。1986 年, E. Rumelhart 和 L. McClelland 提出了一种多层前馈网络的反向传播学习算法^[58],即著名的 Back-Propagation (BP) 算法,这是迄今为止影响最大的网络学习算法,该算法证明了多层神经网络具有很强的学习能力。1988 年, Cybenko 等人从理论上证明了三层神经网络具有以任意精度逼近任意连续非线性函数的能力,即神经网络的网能逼近特性^[59]。这些神经网络理论的提出为神经网络在系统控制中的研究和应用打下了坚实的基础。

在神经网络方法进入控制领域之前,传统的控制律设计方案是建立数学模型,采用数学表达式描述系统,然后进行相应的数学求解。这种方法与数学模型直接相关,模型不精确和参数扰动都会影响控制效果^[66]。

神经网络控制不采用传统的设计思路,无需被控对象的数学模型,具有良好的自适应和自学习能力,借助于神经网络连接权值的自调整,可以放宽对模型精度的依赖。

对于控制学科而言,神经网络的巨大吸引力主要在于^[59]:神经网络本质上是非线性系统,能够充分逼近任意复杂的非线性系统;神经网络具有高度的自适应

性和自组织性，能够学习和适应严重不确定性系统的动态特性；系统信息等势分布存储在网络的各神经元及其连接权中，故有很强的鲁棒性和容错能力；信息的并行处理方式使得快速进行大量运算成为可能。这些优点说明神经网络在解决高度非线性和严重不确定性系统的控制方面具有较大的潜力。目前，基于神经网络的不确定非线性系统自适应控制和容错控制已成为控制理论研究的热点之一，引起了广大学者的关注。

目前将神经网络用于重构控制主要有以下几种方式：①神经网络本质上是非线性系统，能够充分逼近任意复杂系统，应用神经网络建立系统的观测器模型，可以在不知道具体故障信息的情况下，实现对故障环节的隔离和重构，保证飞行的安全性和可靠性；②神经网络具有高度自适应性，采用神经网络单独作为控制器或者和其他算法结合构成自适应控制器，可以保证系统的容错能力；③采用神经网络优化重构算法，提高算法的效率。

随着系统的日趋庞大复杂，神经网络智能容错控制的优越性得到了人们的重视，成为控制领域的研究热点。目前对神经网络的研究主要集中学习算法、结构优化和稳定性的研究上。

由于神经网络的误差函数曲面是多维空间中一个形状极为复杂、不规则的曲面，存在多个局部极小点，使得传统的神经网络学习算法在训练过程中，很容易陷入局部极小点而不能自拔；并且训练神经网络时，网络的参数初始值是在一定范围内随机产生的较小值，在随机产生的初始值下，网络训练从误差曲面的某个初始点开始调整，算法对网络的初始参数异常敏感，不同的初始权值和阈值可能导致完全不同的结果。同时，在神经网络的设计中，神经网络的结构也关系到网络设计的好坏，结构过于简单会导致网络的容错性和自适应性降低，特别是处理较复杂的问题时容易导致神经网络难以训练出理想的结果；而隐含层神经元过多则会增加网络的学习时间，传统的结构设计一般根据经验公式或者实验的方法来确定神经网络的结构，带有很大的盲目性；

另一方面，从神经网络的函数逼近功能这方面划分，神经网络分为全局逼近网络和局部逼近网络^[66]。如果神经网络的一个或者多个可调参数（比如连接权值和阈值）在输入样本空间的每一点处对任何一个输出都有影响，则称这种神经网络为全局逼近网络。全局逼近神经网络考虑了输入样本中的每一个输入，通过连接权的调整来实现对输出的影响，具有良好的逼近能力，但这种神经网络难以满足要求在线控制的实时性要求。相对的，如果对网络输入空间的某个局部区域只有少数连接权影响网络输出，则称这种神经网络为局部逼近网络。局部逼近网络不必对所有的连接权进行训练，只需要对影响网络输出的输入连接权进行训练，学习速度快，可有效的应用于实时控制。

针对不同的应用策略,本文在第五章和第六章分别设计了不同的神经网络学习算法和控制形式,相应的解决了快速有效的学习算法、结构优化、保证重构控制律后闭环系统的稳定性和神经网络收敛性等问题。

粒子群优化算法是一种基于群智能的演化计算技术,源于对鸟群和鱼群捕食行为的启发,具有流程简单、算法简洁、全局优化等特点。利用粒子群算法训练神经网络,网络收敛速度较快^[169]。目前使用粒子群算法并行优化神经网络结构和权值的研究还很少,如何改善粒子群算法的性能,并用于神经网络结构和权值的优化当中,是提高神经网络的学习能力和改善网络结构设计的一个很有潜力的途径。

反步递推(backstepping)设计方法又称后推法、回推法或反演法等,自Kokotovic^[71]等人于1991年提出后就受到了广大学者的重视^[72-77],通常与Lyapunov型自适应律综合使用,使闭环系统达到期望的动静态特性。采用反步递推法训练神经网络容错控制器,可以实现系统的输出镇定和网络学习收敛。

(7) 变结构重构控制

变结构控制是在相平面法的基础上产生的一种现代控制理论的综合方法。从广义上看,目前变结构控制系统主要有两类:一类具有滑动模态的变结构控制系统,称为滑动模变结构控制或滑动模控制;一类是不具有滑动模态的变结构控制。一般变结构控制主要指前者,这是由于具有滑动模态的变结构系统不仅对于系统的不确定性因素具有不变性,而且可以通过设计滑动模态来获得满意的动态品质^[170]。

变结构重构控制是建立在滑模变结构控制理论上,滑模变结构控制对扰动以及对对象不确定性不敏感,所以设计的重构控制律具有很强的鲁棒性,但是这种方法需要保证飞机发生故障时滑模变结构重构控制的不变性。

从研究的动态来看,鲁棒技术、自适应控制和智能控制在重构控制概念下的交叉、融合已经奠定了容错飞行控制的基础,并成为一个最具有应用前景的发展方向。

1.2.3 容错控制研究的若干问题讨论

容错控制在理论和工程上积累了丰富的成果,但也存在很多问题,有待于进一步的研究和讨论,主要表现在:

(一) 故障检测与诊断中存在的问题

故障检测与诊断是主动容错控制的基础,对系统故障的误报或者漏报都可能对控制效果产生不良后果;同时从故障发生到检测出故障存在一定的时延,这段

时延过长有可能会对故障系统的动态性能和稳定性产生严重的影响,因此研究快速、高效的故障检测与诊断方法具有重要的意义。

同时考虑到模型不确定性、外界噪声等因素,人们要求故障诊断机构具有一定的鲁棒性,这和故障诊断机构对故障的灵敏性存在着矛盾。在故障诊断技术的设计中,要综合考虑矛盾的双方选择最优方案。

故障检测与诊断作为一门学科,目前还处于形成和发展之中,在故障诊断的鲁棒性、自适应性和各种诊断方法的综合使用等方面还需进一步的完善。

(二) 控制设计中存在的问题

(1) 重构控制方法问题

重构控制设计保证了故障系统的稳定性和可接受的动态性能,是主动容错控制的核心内容。现有的容错控制方法虽然取得了阶段性的成果,但是也存在着一一些问题,比如,基于特征结构配置的重构控制是容错飞行控制的重要方法之一,然而这种策略对于故障可能引起的维数不匹配情况失效,并且传统的基于输出反馈的特征结构配置重构方法不能配置全部闭环特征值而可能引起系统不稳定,这些缺陷限制了该方法的广泛应用;简单自适应容错控制的设计流程简单,方法固定,适用于飞行系统的实时容错控制,但是这种方法存在对被控对象要求过严和控制律计算困难等不足,如何简化控制器的设计以提高实时容错控制的效率是人们关心的问题。

(2) 重构控制的鲁棒性和自适应性问题

我们除了要求故障诊断机构具有鲁棒性,还要求重构控制律具有鲁棒性和自适应性;同时由于系统的非线性和不确定的存在,导致系统精确模型在很多情况下都无法得到,这为重构控制带来很大的困难。

(3) 非线性系统的容错控制问题

目前非线性系统的容错控制是一个难点,这主要是由于对非线性系统缺乏一般性的控制器综合方法^[80],非线性系统故障诊断问题也不成熟,关于非线性容错控制的研究成果主要集中在智能容错控制方法和线性化容错控制方法。

(4) 分散大系统/关联系统的容错控制问题

现代系统日趋庞大复杂,系统通常包含多个子系统,并且每个子系统间可能存在着状态、输入、输出间的强耦合。相对于集中控制,分散控制系统更容易实现容错控制。因此复杂大系统的容错控制问题也可以被看作是关联系统间的容错控制问题,对这个问题的研究有着重要的理论意义和实用价值。

(5) 时滞系统的容错控制问题

由于对时滞动态系统缺乏有效的故障诊断方法,时滞系统的容错控制进展很小,非线性时滞系统的容错控制还没有任何成果,线性时滞系统容错控制的成果

还非常有限，主要集中在线性定常系统的完整性控制问题。但在工程实际中，很多系统都有时滞性，因此研究时滞动态系统的容错控制具有重要的工程意义，是今后容错控制研究的一个发展方向。

上面讨论了容错控制在理论研究方面存在的问题。在工程实际中，容错控制面临的问题复杂多样，应针对具体问题，考虑不同容错控制方法的特点进行选择。同时系统的数学模型未必可得，或者即使可得也不可避免的和实际系统存在偏差，这也是容错控制遇到的困难之一。因此在工程中，设计人员要根据具体情况设计合适的方案，保证容错控制的效果。

1.3 本文的内容安排

本文主要针对现代飞行系统进行了容错控制研究和设计，内容安排如下：

第一章主要介绍了论文研究的目的和意义、研究现状和发展方向。

第二章主要介绍了飞行控制系统故障诊断和容错控制的基本概念，讨论了飞行器常见故障及其数学模型，并介绍了本文研究的飞控系统自适应容错控制方法和理论。

第三章研究了基于特征结构配置的容错控制方法。该方法的基本思路是故障后通过重新设计特征结构配置反馈增益矩阵，使故障后系统的闭环特征值和特征向量最大程度地恢复到正常状态的特征值和特征向量，从而保证系统的性能得到最大程度的恢复。考虑到故障可能引起的不可观或不可控模态，采用奇异值分解的办法改进了特征结构配置容错控制设计方法；进一步针对故障可能引起的维数不匹配情况，以及传统的基于输出反馈的特征结构配置重构方法仅能配置 $\max(m, p)$ 个特征值而可能引起的系统不稳定性，本章提出一种综合了特征值灵敏度和特征向量的寻优设计指标，在此基础上结合改进的粒子群算法，提出一种特征结构配置重构控制设计方法。

第四章在模型跟随自适应的基础上，基于指令跟踪发生器理论，研究了简单自适应容错控制方法。这种设计方案无需具体的飞行系统故障信息，设计简单，方法固定，适用于飞行系统的实时容错控制。本章分别采用前向补偿器和反馈补偿器保证了系统的严格正实性；在自适应控制器部分，引入误差积分抑制外部扰动，改进了简单自适应控制器的鲁棒性，并进一步提出一种变步长修正的简单自适应容错控制方法，仿真结果表明在满足正实性的前提下，采用简单自适应容错控制，故障系统的输出可跟随理想参考模型的输出。

第五章研究了神经网络在飞行控制系统故障诊断与自适应重构中的一种应

用,提出一种对神经网络的结构和权值并行训练的学习策略,该策略采用爬山思想改进了粒子群算法,提高了粒子群的效率和解的质量;结合蚁群算法的信息素思想调整神经网络隐层结构,充分利用了群体的搜索经验,实现了优化过程中不同结构间信息的共享,从而完成神经网络结构和权值的并行优化。在此基础上,离线建立了神经网络辨识器模型;然后根据飞控系统实时、闭环的特点,结合残差序贯概率比假设检验,采用神经网络辨识器对飞控系统进行故障隔离与自适应重构设计,并对设计的算法进行仿真验证。

第六章研究了基于神经网络控制器的容错飞行控制方法。对于非线性不确定的飞行控制系统故障,本章结合局部逼近神经网络和一种系统化的设计方法-Backstepping 的思想,提出一种自适应鲁棒容错飞行控制方法。容错控制器采用 Backstepping 学习算法,通过虚拟控制量由前向后递推,最终设计了全局容错控制律和神经网络权值调整算法,实现了系统的输出跟踪;该方法中,采用 CMAC 同时学习故障后的不确定性和虚拟控制量的导数信息,避免了 Backstepping 的计算膨胀问题。这种设计策略为控制 Lyapunov 函数的构造提供了一类较为简便的结构化、系统化方法,保证了系统的输出跟踪和神经网络学习收敛。仿真结果表明,容错控制达到了理想的效果。

第七章对全文的研究工作做了总结,并讨论了研究中存在的问题和容错控制的发展趋势。

第二章 飞行控制系统故障诊断与容错控制的基本概念

为了提高飞行的可靠性，现代飞行控制系统一般采用多余度配置，随着系统集成度的提高以及对重量、空间和维修性的考虑，除一些重要的部件外，其余部件余度配置较低，快速高效的故障诊断和重构控制算法为保证系统的高可靠性开辟了新的途径。

本章以飞行控制系统为研究对象，详细描述了飞行控制系统故障诊断与自适应重构的基本概念，分析了常见的故障类型，建立了传感器、作动器等系统部件的故障模型，介绍了飞行控制系统故障诊断与自适应重构技术的一些基本方法，为后续研究打下基础。

2.1 引言

现代飞机系统是多变量时变高阶系统，是一个十分复杂的被控对象，其复杂性主要表现在以下几个方面^[102]：

(1) 多变量

现代飞机是一个多输入/多输出的被控对象，且状态变量很多。比如就操纵面而言，除了具有常规的升降舵、方向舵、副翼这三种操纵面，根据功能和任务的不同，还分别具有鸭翼、缝翼、襟翼等。

(2) 时变性

现代飞机在执行飞行任务时，飞行高度和马赫数是时变的，并且变化范围很大，飞机动压也会改变，最终会引起飞机模型参数的大范围变化。同时在飞行控制系统运行的过程中，工艺设备参数也会不断发生变化，导致系统模型参数发生改变。这些因素都会导致飞机运动特性的时变性。

(3) 时滞、高阶系统

现代飞机六自由度数学模型具有很高的阶次，结合传感器、执行器等特性，系统阶次会更高。在飞行控制系统中，很多环节行为上都表现为纯滞后，纯滞后降低了系统的稳定性，并且会造成开环系统的响应滞后增大，闭环系统的幅值和相位减小，导致整个闭环系统的动态品质下降。

(4) 耦合

飞机的横向运动与航向运动之间、纵向运动和速度之间存在严重的耦合。以横向运动与航向运动的耦合为例，当滚转角速度非零时，会产生一个非零的偏航

角速度；反之偏航角速度非零时，相应会有一个非零的滚转角速度。

除了上面提到的飞机复杂因素以外，还要考虑系统模型的不确定性，飞机系统本身所固有的非线性和执行机构的非线性，飞行环境的各种变化等等，在这种背景下提高飞行控制系统的安全性和可靠性是一项具有重要意义的课题。

2.2 飞机常见故障建模

传感器和作动器是现代飞行控制系统中最重要的组成部分，其可靠性直接关系到系统性能的优劣。基于安全性考虑，传感器在现代飞机中一般采用多余度配置。为减轻飞机重量，除一些重要的传感器系统外，很多传感器余度配置较低。传感器的任何故障都可能导致飞行控制系统需要的反馈信号来源中断，对飞行控制性能产生不利影响，所以传感器的可靠性一直是现代飞机飞行控制系统需要解决的关键问题。作动器负责执行飞行控制系统的操纵指令，其性能直接影响着飞机的飞行品质，为保证电传飞行控制系统的高可靠性，现代先进飞机大都使用多余度的操纵面和伺服机构，但是由于各余度通道之间的相互影响和信号耦合，使得系统的设计非常复杂，所以研究作动器的重构控制以增强其可靠性是非常必要的，在作动器发生故障情况下保证飞机正常飞行，对提高飞机的安全性和生存能力具有重要意义。研究表明系统故障的 60% 以上是发生在控制系统，而控制系统的故障多为传感器和作动器的系统故障，特别是传感器故障更容易发生，本节讨论了几种常见的传感器和作动器故障，并建立了对应的数学模型。

考虑线性定常控制系统：

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) \end{cases} \quad (2-1)$$

式中， $x(t) \in R^n$ 为系统状态变量， $u(t) \in R^m$ 为系统输入， $y(t) \in R^p$ 为系统输出，

$A \in R^{n \times n}$ 为系统矩阵， $B \in R^{n \times m}$ 为系统输入矩阵， $C \in R^{p \times n}$ 为系统输出矩阵。

故障后，系统的状态方程描述为：

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) + E_x f_x(t) \\ y(t) = Cx(t) + E_y f_y(t) \end{cases} \quad (2-2)$$

其中， E_x 和 E_y 为故障分布矩阵， $f_x(t)$ 和 $f_y(t)$ 为故障时间特性标量，分别表示故障对输入的影响函数和对系统输出的影响函数。

（一）作动器故障

当系统发生作动器故障后,系统参数发生变化,状态方程相应将会发生改变。通常情况下,作动器故障主要有作动器卡死、松浮、损伤和饱和四种类型。 u_{if} 表示故障后第 i 个作动器的实际输出, u_{im} 表示正常工作时第 i 个作动器的输出值, u_{\min} 和 u_{\max} 分别为作动器输出允许的最小值和最大值。下面讨论不同情况下的故障参数模型。

(1) 作动器卡死

第 i 个作动器卡死的故障模型描述为:

$$u_{if} = c_i \quad (2-3)$$

上式中, c_i 为常数, 且满足 $u_{\min} \leq c_i \leq u_{\max}$ 。

此时系统的状态方程描述为:

$$\dot{x}(t) = Ax(t) + Bu(t) + B \begin{bmatrix} 0 \\ \dots \\ c_i - u_{im} \\ \dots \\ 0 \end{bmatrix} \quad (2-4)$$

对照式 (2-2), 有 $E_x = Be_i$, $e_i = [0 \ \dots \ 1 \ \dots \ 0]^T$ 即第 i 个元素为 1 的列向量,

$$f_x(t) = c_i - u_{im}(t)。$$

(2) 作动器松浮

作动器松浮是一种特殊的故障类型,表示作动器不受控制,只是随着飞机的飞行呈现飘浮状态。

第 i 个作动器松浮的故障模型描述为:

$$u_{if} = 0 \quad (2-5)$$

此时系统的状态方程描述为:

$$\dot{x}(t) = Ax(t) + Bu(t) + B \begin{bmatrix} 0 \\ \dots \\ -u_{im} \\ \dots \\ 0 \end{bmatrix} \quad (2-6)$$

对照式 (2-2), 有 $E_x = Be_i$, $f_x(t) = -u_{im}(t)。$

(3) 作动器损伤

第 i 个作动器损伤的故障模型描述为：

$$u_{if} = S_i u_{im} \quad (2-7)$$

上式中, s_i 称为损伤系数, 并且当 $s_i = 1$ 时, 第 i 个作动器正常工作; 当 $s_i = 0$ 时, 第 i 个作动器松浮。

此时系统的状态方程描述为：

$$\dot{x}(t) = Ax(t) + Bu(t) + B \begin{bmatrix} 0 \\ \dots \\ (S_i - 1)u_{im} \\ \dots \\ 0 \end{bmatrix} \quad (2-8)$$

即 $E_x = Be_i$, $f_x(t) = (s_i - 1)u_{im}(t)$ 。

(4) 作动器饱和

作动器饱和即作动器开路失效, 表示作动器停留在输出的最小状态或者最大状态。

第 i 个作动器饱和的故障模型描述为：

$$u_{if} = u_{\min} \quad (2-9)$$

或者

$$u_{if} = u_{\max} \quad (2-10)$$

此时系统的状态方程描述为：

$$\dot{x}(t) = Ax(t) + Bu(t) + B \begin{bmatrix} 0 \\ \dots \\ u_{i\min} - u_{im} \\ \dots \\ 0 \end{bmatrix} \quad (2-11)$$

对照式 (2-2), 有 $E_x = Be_i$, $f_x(t) = u_{i\min} - u_{im}(t)$ 。

或者此时系统的状态方程描述为：

$$\dot{x}(t) = Ax(t) + Bu(t) + B \begin{bmatrix} 0 \\ \dots \\ u_{i\max} - u_{im} \\ \dots \\ 0 \end{bmatrix} \quad (2-12)$$

对照式 (2-2), 有 $E_x = Be_i$, $f_x(t) = u_{i\max} - u_{im}(t)$ 。

(5) 作动器恒偏差

第 i 个作动器恒偏差的故障模型描述为:

$$u_{if} = u_{im} + a_i \quad (2-13)$$

上式中 a_i 为常数。

此时系统的状态方程描述为:

$$\dot{x}(t) = Ax(t) + Bu(t) + B \begin{bmatrix} 0 \\ \dots \\ a_i \\ \dots \\ 0 \end{bmatrix} \quad (2-14)$$

对照式 (2-2), 有 $E_x = Be_i$, $f_x(t) = a_i$ 。

(二) 传感器故障

常见的传感器故障行为有卡死、增益变化、恒偏差三种, 下面分别描述其故障发生时系统的模型。 y_{if} 表示故障后第 i 个传感器的实际输出, y_{im} 表示正常工作第 i 个传感器的输出值。

(1) 传感器卡死

第 i 个传感器卡死的故障模型可描述为:

$$y_{if} = c_i \quad (2-15)$$

上式中, c_i 为常数。

系统的输出方程描述为:

$$y(t) = Cx(t) + e_i \begin{bmatrix} 0 \\ \dots \\ c_i - y_{im} \\ \dots \\ 0 \end{bmatrix} \quad (2-16)$$

对照式 (2-2), $E_y = e_i$, e_i 为第 $i \times i$ 个元素为 1 的方阵, $f_y(t) = c_i - y_{im}(t)$ 。

(2) 传感器增益变化

第 i 个传感器增益变化的故障模型描述为:

$$y_{if} = s_i y_{im} \quad (2-17)$$

上式中, s_i 为损伤系数, 并且当 $s_i = 1$ 时, 第 i 个传感器正常工作。

系统的输出方程描述为：

$$y(t) = Cx(t) + e_i \begin{bmatrix} 0 \\ \dots \\ (s_i - 1)y_{im} \\ \dots \\ 0 \end{bmatrix} \quad (2-18)$$

对照式 (2-2), $E_y = e_i$, $f_y(t) = (s_i - 1)y_{im}(t)$ 。

(3) 传感器恒偏差

第 i 个传感器恒偏差的故障模型可描述为：

$$y_{if} = y_{im} + a_i \quad (2-19)$$

上式中, a_i 为常数。

系统的输出方程描述为：

$$y(t) = Cx(t) + e_i \begin{bmatrix} 0 \\ \dots \\ a_i \\ \dots \\ 0 \end{bmatrix} \quad (2-20)$$

对照式 (2-2), $E_y = e_i$, $f_y(t) = a_i$ 。

(三) 系统部件故障一般情形

按照时间特性的不同, 系统故障可分为突变性故障和缓变故障, 系统部件故障一般表现为系统矩阵 A 和 B 的变化。

(1) 系统发生突变性故障

突变性故障指参数突然出现很大偏差, 事先不可监测和预测的故障。当系统发生突变性故障时, 系统矩阵为：

$$A_f = A + \mathbf{V}A, \quad B_f = B + \mathbf{V}B。$$

此时系统的状态方程可描述为：

$$\begin{aligned} \dot{x}(t) &= (A + \mathbf{V}A)x(t) + (B + \mathbf{V}B)u(t) \\ &= Ax(t) + Bu(t) + [\mathbf{V}A \quad \mathbf{V}B] \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} \end{aligned} \quad (2-21)$$

对照式 (2-2), 即 $E_x = [\mathbf{V}A \quad \mathbf{V}B]$, $f_x = \begin{bmatrix} x(t) \\ u(t) \end{bmatrix}$ 。

(2) 系统发生缓变故障

缓变故障又称为软故障，指参数随时间的推移和环境的变化而缓慢变化的故障。当系统发生缓变故障即参数漂移时，矩阵 A 和 B 不再是常数矩阵，此时系统的状态方程可表示为：

$$\dot{x}(t) = A(r)x(t) + B(r)u(t) \quad (2-22)$$

其中 $r = [r_1 \ r_2 \ \dots \ r_l]$ 是不确定性参数。

系统正常工作状态下的参数为 r_0 ，那么式 (2-22) 在 r_0 处的一阶泰勒展开式为：

$$\begin{aligned} \dot{x}(t) &= A(r_0)x(t) + B(r_0)u(t) + \sum_{i=1}^l \left[\frac{\partial A}{\partial a_i} \mathbf{V}a_i x(t) + \frac{\partial B}{\partial a_i} \mathbf{V}a_i u(t) \right] \\ &= A(r_0)x(t) + B(r_0)u(t) + \begin{bmatrix} \frac{\partial A}{\partial a_1} & \frac{\partial B}{\partial a_1} & \dots & \frac{\partial A}{\partial a_l} & \frac{\partial B}{\partial a_l} \end{bmatrix} \begin{bmatrix} \mathbf{V}a_1 x(t) \\ \mathbf{V}a_1 u(t) \\ \dots \\ \mathbf{V}a_l x(t) \\ \mathbf{V}a_l u(t) \end{bmatrix} \end{aligned} \quad (2-23)$$

将式 (2-23) 对照式 (2-2)，即有 $E_x = \begin{bmatrix} \frac{\partial A}{\partial a_1} & \frac{\partial B}{\partial a_1} & \dots & \frac{\partial A}{\partial a_l} & \frac{\partial B}{\partial a_l} \end{bmatrix}$ ，

$$f_x = [\mathbf{V}a_1 x^T(t) \ \mathbf{V}a_1 u^T(t) \ \dots \ \mathbf{V}a_l x^T(t) \ \mathbf{V}a_l u^T(t)]^T。$$

由上面的分析可以看出，不同的系统故障模型之间可以相互转化，这给故障检测和诊断的研究带来了很大的便利^[17]。

2.3 故障检测与诊断技术

故障诊断技术发展过程中，在不同的应用领域，与不同的学科相结合产生了多种故障诊断方法。概括地讲，现有的故障诊断方法可以分成三大类：基于模型的故障诊断方法、基于信号处理的故障诊断方法和基于知识的故障诊断方法，如下图所示。

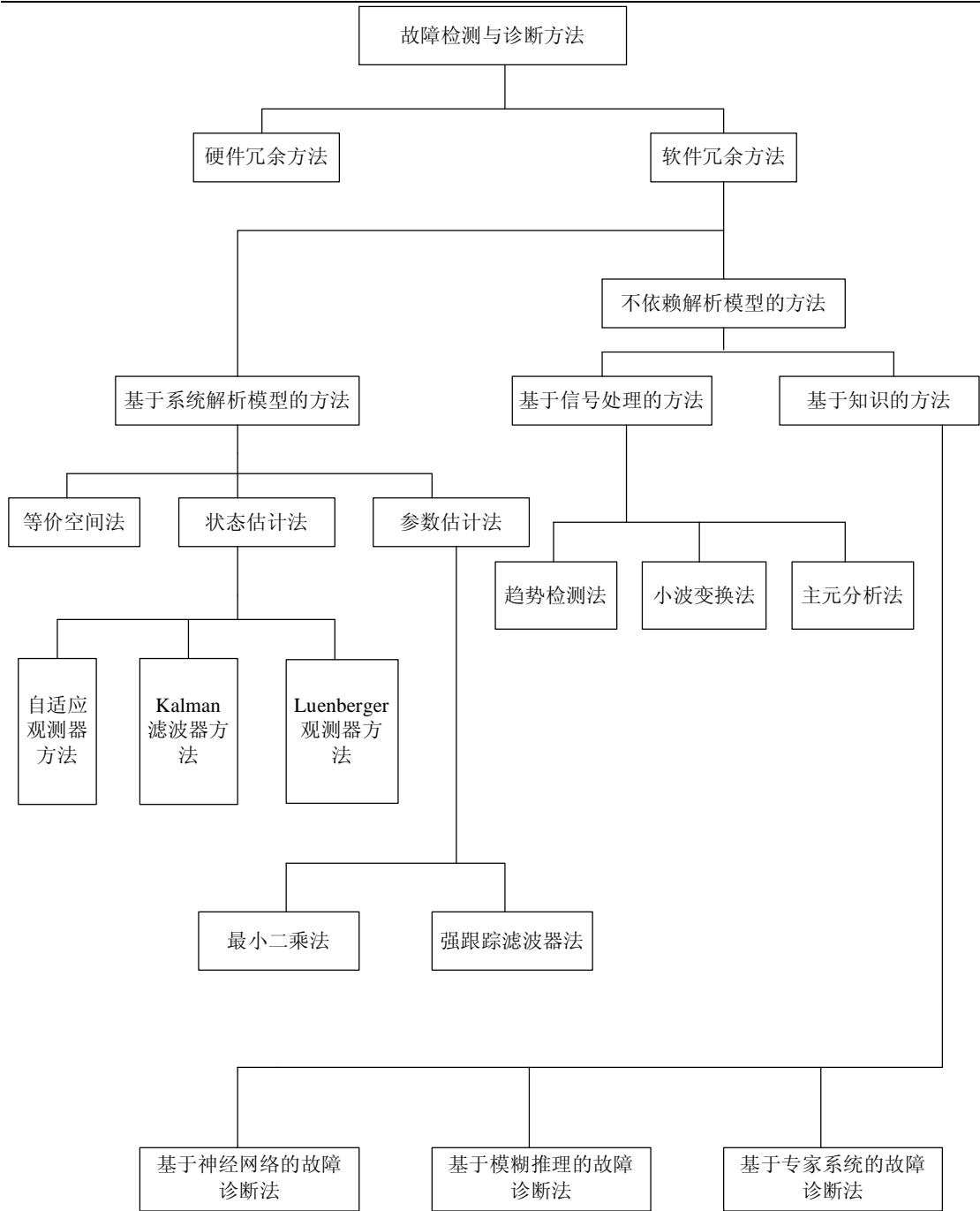


图 2-1 故障诊断方法分类

评价故障诊断系统的指标主要包括检测性能指标、诊断性能指标和综合性能指标^[26,27,42]。

(一) 检测性能指标

(1) 早期检测的灵敏度：早期检测的灵敏度是指一个故障检测系统对小故障信号的检测能力。检测系统早期检测的灵敏度越高，表明它能检测到的最小故障信号越小。

(2) 故障检测的及时性：故障检测的及时性是指发生故障后，检测系统在尽可能短的时间内检测到故障发生的能力。故障检测的及时性越好，说明从故障

发生到故障被正确检测出来之间的时间间隔越短。

(3) 故障的误报率和漏报率：误报是指系统没有发生故障却被错误地判定出现了故障；漏报则是指系统中出现了故障却没有被检测出来的情形。一个可靠的故障检测系统应当保持尽可能低的误报率和漏报率。

(二) 诊断性能指标

(1) 故障分离能力：故障分离能力是指诊断系统对不同故障的区分能力。分离能力的强弱取决于对象的物理特性、故障大小、噪声、干扰、建模误差以及诊断算法。分离能力越强，表明诊断系统对于不同故障的区分能力越强，对故障的定位越准确。

(2) 故障辨识的准确性：故障辨识的准确性是指诊断系统对故障的大小及其时变特性估计的准确程度。故障辨识的准确性越高，表明诊断系统对故障的估计就越准确，越有利于故障的评价与决策。

(三) 综合性能指标

(1) 故障诊断系统的鲁棒性：故障诊断系统的鲁棒性是指故障诊断系统在存在噪声、干扰、建模误差的情况下正确完成故障诊断任务，同时保持理想误报率和漏报率的能力。故障诊断系统的鲁棒性越强，表明其受噪声、干扰、建模误差的影响越小，故障诊断的可靠性越高。

(2) 故障诊断系统的自适应能力：故障诊断系统的自适应能力是指故障诊断系统对于被诊断对象具有自适应能力，能够充分利用变化产生的新信息来调整自身。引起被诊断对象变化的原因可能是工作条件的变化，包括被诊断对象外部输入的变化、结构的变化等等；也可能是故障发生导致的系统变化，因此自适应能力也包括了连续故障检测的能力。

在工程设计中，通过分析实际条件和设计目标，以及设计的主要性能和次要性能，结合检测性能、诊断性能以及综合性能这三个故障诊断系统性能的评判标准，对众多的故障诊断方法进行分析和取舍，最终选定理想的故障诊断方案。

2.4 飞行控制系统的容错控制方法

容错飞行控制的目的是提高飞行器的安全性，在飞机发生故障及损伤的情况下，可以弥补故障带来的飞机动态特性的变化，保证飞行的安全和任务的完成。

本文研究的容错控制方法主要有：

(1) 基于特征结构配置的容错飞行控制设计

线性系统容错控制中通过设计一个新的反馈增益矩阵实现重构是一个很重要的内容，目前主要有三种方法：LQR、广义逆法、特征结构配置法^[17]。基于

LQR 的容错控制一般可以保证故障后闭环稳定性，并可以保持一定的鲁棒性；但是由于 LQR 优化过程中权矩阵的选取概念模糊，具有很大的不确定性和任意性；广义逆法的基本思想是通过设计一个新的反馈增益矩阵，使得重构系统和正常系统的闭环系统矩阵之差达到 Frobenius 范数最小，从而保证重构后的系统尽可能地与正常系统相似，但是广义逆法不能保证重构后闭环系统的稳定性；90 年代中期，学者们提出使用特征结构配置方法对故障系统进行重构控制设计^[29]。文[156]是最早研究特征结构配置重构控制系统设计方法的文章之一，通过对系统的特征值和特征向量最大程度的恢复从而完成对故障系统的重构设计。文献[155,176,177]研究了含约束的输出反馈特征结构配置方法在飞行控制系统设计中的应用，指出含约束的输出反馈设计可以降低飞行控制系统的复杂性并提高其可靠性。文[154,157]针对飞机发生气动失效故障而导致故障前后系统阶次不匹配的情况下传统的特征结构配置设计算法失效的情况，通过分析系统的输出响应与系统特征结构之间的关系，得到了重构设计的优化指标，采用遗传算法设计反馈增益矩阵，使得闭环系统的特征结构得到了最大程度的恢复。

从工程实践角度来讲，基于全状态反馈的特征结构配置重构算法具有一定的局限性，基于输出反馈控制的特征结构配置重构算法具有更大的实用价值。基于输出反馈的特征结构配置方法虽然在飞行控制系统和重构飞行控制系统的设计中取得了一定的应用，但也表现在出两个突出的问题需要改进和解决：（1）闭环系统的稳定性问题。基于输出反馈的特征结构配置不能配置全部闭环特征值，已有的研究成果中^[176,177]通常是在设计完成后进行闭环系统的特征值检测，把不能保证闭环系统稳定的解排除，这种方法仅是一种后验性的试凑方法，缺乏严格的数学理论推导和证明；或者使用一个 Lyapunov 方程的方法保证了重构后闭环系统的稳定性^[178]，并把该方程作为优化指标加入到重构设计指标中，但这种方法的设计过程较为复杂。（2）传统的含约束的输出反馈的特征结构配置方法仅能得到一个最小二乘意义下的解，导致配置后闭环系统的特征值与期望特征值之间存在较大的误差。而配置特征值的精确性问题在重构飞行控制系统的设计是尤为重要，重构设计的目的之一是使闭环系统的极点得到最大程度的恢复。这些问题有待于进一步的研究。

（2）基于简单自适应控制的容错飞行控制设计

基于特征结构配置的重构设计方法要求知道系统的精确故障模型，根据故障信息设计相应的重构控制律以保证飞行品质。而基于模型参考自适应方法的重构控制律设计不需要了解具体的故障信息，只关心理想的飞行品质要求，对故障系统进行容错控制律设计，使得故障系统的输出跟踪参考模型的输出以达到设计要求。本文在模型参考自适应理论的基础上，进一步研究了简单自适应容错控制。

简单自适应控制于 1979 年由 H.Kaufman, K.Sobel 等学者提出^[30], 由于算法简单、设计方便, 在工程中得到了广泛应用。简单自适应控制作为自适应控制的分支, 与常规的模型参考自适应控制的设计目标相同, 都是根据性能品质, 选定理想的参考模型, 通过自适应算法使得被控系统的输出特性与参考模型的输出特性保持一致; 与常规模型参考自适应理论不同的是, 简单自适应控制是建立在指令跟踪发生器或输出渐进跟踪理论上的自适应控制方法, 这种算法充分调动了参考模型的信息, 利用参考模型的输入信息、输出信息以及系统输出和参考模型输出之间的误差来调节自适应控制器参数, 使得系统自动的跟随参考模型的输出。在自适应控制器设计中, 在满足系统稳定性条件的前提下, 只需给出参考增益矩阵的初值, 就可以使得自适应机制自动完成控制器的设计。

为了保证简单自适应控制算法的收敛性和控制系统的稳定性, 要求被控对象满足几乎严格正实性条件 (ASPR), ASPR 条件比较苛刻, 很多实际系统并不满足, 对此国内外学者做了相应的研究工作。本文在第四章分别设计了前馈环节和反馈环节构建扩展系统的方法以满足 ASPR^[44,45], 在此基础上结合优化算法, 进一步提高了算法的质量, 保证了容错控制的需求; 并进一步的研究了自适应容错控制器部分的设计。

(3) 基于神经网络的容错飞行控制设计

目前多种容错控制技术得到研究和发展, 并在工程应用中取得了阶段性研究成果。但是绝大多数的容错控制方法都要求精确的故障模型和故障参数信息, 这一缺点大大限制了该方法在容错控制中的应用; 同时由于故障可能导致出现的系统非线性, 使得基于线性模型和控制理论的重构设计方法失效。神经网络是对人脑或者自然神经系统若干特性的抽象和模拟, 包含了大脑神经的许多特征, 通过自学习, 神经网络在理论上可以以任意精度逼近学习对象, 具有高度的自适应性和自组织性, 能够学习和适应严重不确定性系统的动态特性, 并且系统信息等势分布存储在网络的各神经元及其连接权中, 因此有很强的鲁棒性和容错能力; 这些优点使神经网络成为容错控制领域研究的热点, 引起了广大学者的关注。本文分别采用神经网络辨识器和神经网络控制器设计了容错飞行控制系统。

基于神经网络的辨识方法就是将系统辨识问题转化为对某个非线性函数的逼近问题。其基本思想是采用神经网络逼近被测系统, 利用神经网络具有以任意精度逼近任意非线性函数的能力, 通过学习被测系统的输入/输出, 使得系统输出和网络输出的误差达到最小, 从而实现对系统的辨识。神经网络训练的过程即对系统的辨识过程, 训练好的神经网络在外部特性上与被测系统等价。学习结束后, 神经网络模型与实际系统具有相同的输入输出映射特性, 因此可以作为系统的一个物理冗余通道, 用于系统的故障诊断和重构。

与传统的观测器方法相比较,神经网络辨识器用于系统故障隔离与重构控制具有以下几个特点:

①神经网络具有建模简单、逼近能力强的优点,非常适用于复杂、非线性、具有不确定性以及建模困难的系统辨识问题,并且不需要系统的详细特征,只要求系统的输入输出数据,放宽了建立观测器的限制条件。

②使用神经网络建立的辨识器模型是非算式的,是通过网络自组织、自学习和自调整能实现对系统任意精度的逼近,模型信息反映在网络内部的结构、参数和网络权值上,避免了传统方法中建立观测器时的大量解算。

③在加入噪声影响情况下,神经网络辨识器仍能表显出优良的逼近能力,可对系统实现良好跟踪。

对于这种容错控制策略,研究的热点集中在神经网络的学习算法和结构优化上。

而基于神经网络的容错控制方法不采用传统的设计思路,无需被控对象的数学模型,具有良好的自适应和自学习能力,神经网络的训练过程也就是容错控制器的调整、完善过程。神经网络容错控制器的设计主要分为两种方式:一种是采用神经网络单独构成控制器;一种是与其他控制方法结合构成控制器。这种容错控制策略对实时性要求较高,并且系统的稳定性和神经网络的收敛性是人们关心的重点。

根据神经网络容错控制设计方式的不同,采用的神经网络和网络学习算法根据具体问题而不同,神经网络容错控制的理论将愈加丰富。

随着系统的日趋庞大复杂,神经网络智能容错控制的优越性得到了人们的重视,成为控制领域的研究热点。

2.5 小结

本章主要介绍了飞行控制系统故障诊断与自适应容错的基本概念和相关内容,首先分析了传感器、作动器和系统一般部件故障的类型,并建立了常见故障的数学模型,然后介绍了飞行控制系统故障诊断的主要方法及其评判标准,最后综述了本文主要研究的自适应容错飞行控制方法,为后续研究奠定基础。

第三章 基于特征结构配置的容错飞行控制设计

基于特征结构配置的重构方法是容错飞行控制系统研究的重要内容。对线性系统进行响应分析,结果表明闭环特征值决定了系统的稳定性和响应速度,特征向量决定了各运动模态和响应的耦合关系和耦合程度,通过对系统进行特征结构配置可以更好的把握系统的性能;基于特征结构配置重构控制设计的基本思想是对故障后的系统设计特征结构配置反馈增益矩阵,使得故障系统的闭环特征值和特征向量最大程度恢复到标称特征值和特征向量,从而保证重构控制的有效性。基于特征结构配置的重构方法是线性系统容错控制方法中重要的一种,本章根据飞行控制系统的特点,对基于特征结构配置的重构控制进行了深入的研究。

3.1 引言

在线性系统容错控制中,通过设计一个新的反馈增益矩阵实现重构是一个很重要的内容,目前主要有三种方法:线性二次最优调节器法 LQR、广义逆法、特征结构配置法^[17]。基于 LQR 的容错控制一般可以保证故障后闭环稳定性,并可以保持一定的鲁棒性;但是由于 LQR 优化过程中权矩阵的选取概念模糊,具有很大的不确定性和任意性,因此 LQR 难以使故障系统恢复到正常工作状态下的性能。广义逆法的基本思想是通过设计一个新的反馈增益矩阵,使得重构系统和正常系统的闭环系统矩阵之差达到 Frobenius 范数最小,从而保证持重构后的系统尽可能地与正常系统相似。但是广义逆法不能保证重构后闭环系统的稳定性, Gao Z^[28]对广义逆法进行了改进,提出 MPIM 法来保证重构系统的稳定性,但这种改进算法的稳定性约束条件很严格,导致 MPIM 只适用于有限的一类过程扰动,对于其他故障情况,尤其对高维系统无法使用。90 年代中期,学者们提出使用特征结构配置方法对故障系统进行重构控制设计^[29]。相对于极点配置,特征结构配置不但配置闭环系统的极点,而且还配置它们的重数并同时配置闭环特征向量,因而可以更加准确地掌握系统的性能。

极点即线性系统的特征值,由控制系统稳定性和响应分析理论可知,它决定了系统的稳定性和响应速度。基于状态空间理论的极点配置设计是时域反馈控制系统设计的常用方法,其目的是对于某线性定常系统,通过设计适当的状态反馈控制律,使闭环系统具有理想的特征值,从而实现对系统的性能优化。相较于传

统的极点配置, 特征结构配置方法不仅配置了系统的理想特征值, 同时还配置了期望特征向量, 从而可以更加准确地把握系统性能。

考虑线性时不变连续系统, 其状态方程为:

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \\ \mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} \end{cases} \quad (3-1)$$

式中, $\mathbf{x} \in R^n$ 为系统状态变量, $\mathbf{u} \in R^m$ 为系统输入, $\mathbf{y} \in R^p$ 为系统可测量输出, $\mathbf{A} \in R^{n \times n}$ 为系统矩阵, $\mathbf{B} \in R^{n \times m}$ 为系统输入矩阵, $\mathbf{C} \in R^{p \times n}$ 为系统输出矩阵, $\mathbf{D} \in R^{p \times m}$ 为系统控制输出矩阵。在描述一个线性系统的上述矩阵中, \mathbf{A} 矩阵最为重要, 它描述了系统本身所具有的动态特性, 矩阵 \mathbf{B} 描述了系统外部输入对状态 \mathbf{x} 的影响, 矩阵 \mathbf{C} 决定了哪些状态 \mathbf{x} 作为可测输出 \mathbf{y} , 矩阵 \mathbf{D} 描述了外部输入对可测输出 \mathbf{y} 的影响。

矩阵 \mathbf{A} 可以进一步分解为两个要素: 特征值和特征向量, 二者构成了系统矩阵 \mathbf{A} 的特征结构, 其关系可以描述为:

$$\mathbf{A}\mathbf{V} = \mathbf{V}\mathbf{\Lambda} \quad (3-2)$$

式中, $\mathbf{\Lambda} = [\lambda_1 \mathbf{I} \lambda_2 \mathbf{I} \dots \lambda_n \mathbf{I}]$ 为特征值, $\mathbf{V} = [\mathbf{v}_1 \mathbf{I} \mathbf{v}_2 \mathbf{I} \dots \mathbf{v}_n \mathbf{I}]$ 为对应的特征向量。

特征向量 \mathbf{V} 是系统状态的基空间, 线性系统任意时刻的状态都是特征向量的线性组合, 特征向量 \mathbf{V} 称为系统的右特征向量。系统的左特征向量为第二种基空间, 可以描述为:

$$\mathbf{W}\mathbf{A} = \mathbf{\Lambda}\mathbf{W} \quad (3-3)$$

式中, $\mathbf{W}^T = [\mathbf{w}_1 \mathbf{I} \mathbf{w}_2 \mathbf{I} \dots \mathbf{w}_n \mathbf{I}]$ 为系统左特征向量, $\mathbf{\Lambda} = [\lambda_1 \mathbf{I} \lambda_2 \mathbf{I} \dots \lambda_n \mathbf{I}]$ 为特征值。

通过求解 (3-1) 式所描述的线性系统方程, 可以建立起系统状态时间响应、输出时间响应与系统特征值、特征向量之间的关系:

$$\mathbf{x}(t) = \underbrace{\sum_{i=1}^n \mathbf{v}_i \mathbf{w}_i^T e^{\lambda_i t} \mathbf{x}_0}_{\text{自由响应}} + \underbrace{\sum_{i=1}^n \mathbf{v}_i \mathbf{w}_i^T \int_0^t e^{\lambda_i(t-\tau)} \mathbf{B}\mathbf{u}(\tau) d\tau}_{\text{强迫响应}} \quad (3-4)$$

$$\mathbf{y}(t) = \underbrace{\sum_{i=1}^n \mathbf{C}\mathbf{v}_i \mathbf{w}_i^T e^{\lambda_i t} \mathbf{x}_0}_{\text{自由响应}} + \underbrace{\sum_{i=1}^n \mathbf{C}\mathbf{v}_i \mathbf{w}_i^T \int_0^t e^{\lambda_i(t-\tau)} \mathbf{B}\mathbf{u}(\tau) d\tau}_{\text{强迫响应}} \quad (3-5)$$

由 (3-4) 和 (3-5) 式可以看到, 系统状态或输出响应有两部分组成。第一部分与系统初始状态有关, 称为自由响应; 第二部分与系统外部输入有关, 称为

强迫响应。

由此可见，整个线性系统的时间响应由以下四个要素来决定：

- (1) 系统的特征值 λ_i ，决定了系统的稳定性和响应速度；
- (2) 系统的特征向量 v_i, w_i ，决定了各模态间的耦合关系和耦合程度；
- (3) 系统的初始状态 x_0 ；
- (4) 系统的外部输入 $u(t)$ 。

特征结构配置通过合理配置闭环系统的特征值和特征向量，可以更加准确地掌握系统的性能。使用特征结构配置法对飞机故障系统进行重构控制器设计，以使故障后闭环系统的特征值和特征值所对应的特征向量得到最优恢复，从而使故障系统的性能最大限度接近于标称系统。

3.2 基于特征结构配置的重构控制

3.2.1 问题描述

考虑正常情况下的线性多变量连续系统，其状态方程为：

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases} \quad (3-6)$$

式中， $x \in R^n$ 为系统状态变量， $u \in R^m$ 为系统输入， $y \in R^p$ 为系统可测量输出，

$A \in R^{n \times n}$ 为系统矩阵， $B \in R^{n \times m}$ 为系统输入矩阵， $C \in R^{p \times n}$ 为系统输出矩阵，

$D \in R^{p \times m}$ 为系统控制输出矩阵。闭环系统的特征值为 $\{\lambda_i^d\} \ i=1,2,\dots,n$ ，相应的特征向量为 $\{v_i^d\} \ i=1,2,\dots,n$ 。

故障后系统的状态方程描述为：

$$\begin{cases} \dot{x}_f = A_f x_f + B_f u_f \\ y_f = C_f x_f + D_f u_f \end{cases} \quad (3-7)$$

从工程实践角度来讲，由于实际系统中并非所有状态可测，或者状态可测但测量成本较高，因此基于全状态反馈的特征结构配置重构算法具有一定的局限性，而仅仅通过反馈可测量或易测量的变量进行控制无疑具有较大的吸引力，即是输出反馈控制。

对故障后系统采用如下形式的输出反馈控制律：

$$u_f = Ky_f \quad (3-8)$$

因此，特征结构配置重构控制所要解决的问题可以描述为：

给定标称系统的特征结构：特征值-自共轭标量集 $\{l_i^d\} \ i=1,2,\mathbf{L},r$ ，和对应的特征向量-自共轭 n 维向量集 $\{v_i^d\} \ i=1,2,\mathbf{L},r$ ，要找出一个实数矩阵 K ，使得 $A_f + B_f K C_f$ 的特征值和特征向量最大程度地恢复到正常系统，即 $\{l_i^d\} \ i=1,2,\mathbf{L},r$ 和对应的特征向量 $\{v_i^d\} \ i=1,2,\mathbf{L},r$ 。

假设 G （即 $A_f + B_f K C_f$ ）具有 n 个互异特征值 $l_i, i=1,2,\mathbf{L},n$ ，对应的特征向量为 v_i ，则有

$$Gv_i = l_i v_i \quad (3-9)$$

1976 年，Woonham 得出以下定理阐述输出反馈问题的解^[55]。

【定理 3-1】式（3-7）所描述的系统可控，当且仅当对于任意自共轭标量集 $\{l_i^d\} \ i=1,2,\mathbf{L},n$ ，存在实矩阵 K ，使得 $A_f + B_f K C_f$ 的特征向量为 $\{v_i^d\}, i=1,2,\mathbf{L},n$ 。

【定理 3-2】令 $\{l_i\}_{i=1}^n$ 为一组自共轭的互异复数特征值，则存在一个实矩阵 K ，使得 $A_f + B_f K C_f$ 的特征系统方程成立，即

$$(A_f + B_f K C_f)v_i = l_i v_i \quad (i=1,2,\mathbf{L},n) \quad (3-10)$$

当且仅当对每个 i 有，

(1) $\{v_i\}_{i=1}^n$ 为复域 C^n 一组线性独立向量；

(2) $v_i = v_j^*$ 当 $l_i = l_j^*$ 时；

(3) $v_i \in \text{span}\{N_{l_i}\}$

并且，当 K 存在且 $\text{rank} B_f = m$ 时，反馈阵 K 唯一^[37]。

Moore 从本质上描述了所有能够达到给定互异特征值集的闭环特征向量集合所具有的特征。他还指出，除了预先指定的闭环特征值外，还可选择一组适当的特征向量。

Davison 首先从特征值配置方面研究了输出反馈控制问题，研究表明，如果系统可控可观测，且有 $\text{rank}(B_f) = m, \text{rank}(C_f) = p$ ，则可任意配置 $\max(m, p)$ 个特征值。针对输出反馈进行特征结构配置问题，Srinathkumar、Shapiro、Chung 等人进行了研究，取得了重要成果。在 Srinathkumar 的论文中，我们得知如下定理：

【定理 3-3】^[55] 对于式 (3-7) 描述的系统，假设故障不改变其可观和可控性，且矩阵 B_f, C_f 满秩，那么通过输出增益反馈即控制律 $u_f = Ky_f$ ，可配置 $\max(m, p)$ 个闭环特征值，并可局部配置 $\max(m, p)$ 个特征向量，其中每个特征向量中有 $\min(m, p)$ 个元素可任意配置。

通常假设 $p > m$ ，即输出变量维数大于输入变量维数，这种情况在工程实际中经常遇到，特别是在飞行控制问题中。

由此，我们得到下面的引理。

【引理 3-1】对于闭环系统，总存在一个输出反馈增益矩阵 K ，可任意配置系统的 p 个特征值，如果特征值为复数，只需满足共轭约束条件即可。

3.2.2 特征结构配置重构算法实现

相应上述对于故障系统的假定以及关于 $\{I_i^d\}$ 、 $\{v_i^d\}$ 的问题陈述，下面分析可配置的闭环特征向量所具有的特征。

首先，考虑闭环系统

$$\dot{x} = (A_f + B_f K C_f) x \quad (3-11)$$

假设给定的理想闭环系统特征值为 $\{I_i\}_{i=1}^r$ ， v_i 为 I_i 对应的闭环特征向量，对于一组特征值-特征向量 I_i 和 v_i ，有

$$(A_f + B_f K C_f) v_i = I_i v_i \quad (3-12)$$

即

$$v_i = (I_i I - A_f)^{-1} B_f K C_f v_i \quad (3-13)$$

假定闭环系统理想特征值与开环 A 阵特征值不等，即 $(I_i I - A)$ 的逆存在，分

析式 (3-11)，定义 m 维向量 \mathbf{m}_i 如下：

$$\mathbf{m}_i = \mathbf{K}\mathbf{C}_f \mathbf{v}_i \quad (3-14)$$

则式 (3-11) 可写为

$$\mathbf{v}_i = (\mathbf{I}_i \mathbf{I} - \mathbf{A}_f)^{-1} \mathbf{B}_f \mathbf{m}_i \quad (3-15)$$

可知，理想特征向量 \mathbf{v}_i 必须属于 $(\mathbf{I}_i \mathbf{I} - \mathbf{A}_f)^{-1} \mathbf{B}_f$ 的诸列张成的子空间。由于故障，系统的动态性能发生改变，故障后系统的特征向量和标称系统的特征向量 \mathbf{v}_i^d 位于不同的子空间中，以致于理想特征向量不可达。在这种情况下，需求最优可达的特征向量来代替理想特征向量 \mathbf{v}_i^d ，这个最优向量即 \mathbf{v}_i^d 在 $(\mathbf{I}_i \mathbf{I} - \mathbf{A})^{-1} \mathbf{B}$ 的诸列张成的子空间内的投影，记作 \mathbf{v}_i^a ，其示意图如下：

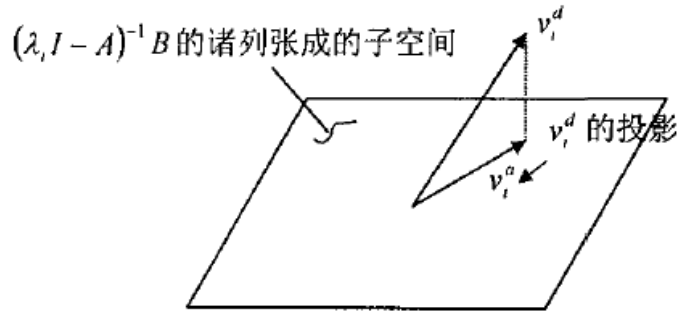


图 3-1 可达特征向量 \mathbf{v}_i^a 的几何解释示意图

下面研究如何计算 \mathbf{v}_i^a 。首先定义向量 \mathbf{L}_i ：

$$\mathbf{L}_i = (\mathbf{I}_i \mathbf{I} - \mathbf{A}_f)^{-1} \mathbf{B}_f \quad (3-16)$$

由前文分析已知，最优可达特征向量 \mathbf{v}_i^a 必须属于 $(\mathbf{I}_i \mathbf{I} - \mathbf{A}_f)^{-1} \mathbf{B}_f$ 的诸列张成的子空间，则有：

$$\mathbf{v}_i^a = \mathbf{L}_i \mathbf{z}_i = (\mathbf{I}_i \mathbf{I} - \mathbf{A}_f)^{-1} \mathbf{B}_f \mathbf{z}_i, \mathbf{z}_i \in \mathbf{R}^m \quad (3-17)$$

选择的 \mathbf{z}_i 要最小化性能指标 J ：

$$\min J = \min \|\mathbf{v}_i^d - \mathbf{v}_i^a\|^2 = \min \|\mathbf{v}_i^d - \mathbf{L}_i \mathbf{z}_i\|^2 \quad (3-18)$$

根据标量函数对向量与矩阵的微分运算，有：

$$\partial J / \partial \mathbf{z}_i = 2 \mathbf{L}_i^T (\mathbf{L}_i \mathbf{z}_i - \mathbf{v}_i^d) \quad (3-19)$$

若令 $\partial J / \partial \mathbf{z}_i = \mathbf{0}$ ，则有

$$\mathbf{z}_i = (\mathbf{L}_i^T \mathbf{L}_i)^{-1} \mathbf{L}_i^T \mathbf{v}_i^d \quad (3-20)$$

$$\mathbf{v}_i^a = \mathbf{L}_i (\mathbf{L}_i^T \mathbf{L}_i)^{-1} \mathbf{L}_i^T \mathbf{v}_i^d \quad (3-21)$$

特征结构配置重构算法中反馈形式为：

$$\mathbf{u}_f = \mathbf{K} \mathbf{y}_f \quad (3-22)$$

系统矩阵为：

$$\mathbf{G} = \mathbf{A}_f + \mathbf{B}_f \mathbf{K} \mathbf{C}_f \quad (3-23)$$

为了便于计算，引入一个线性变换矩阵 \mathbf{T} ， $\mathbf{T} = [\mathbf{B}_f \mid \mathbf{P}]$ ，其中 \mathbf{P} 是可以保证矩阵 \mathbf{T} 满秩的任意矩阵，使用 \mathbf{T} 矩阵，对系统进行线性变换：

$$\begin{cases} \dot{\bar{\mathbf{x}}}_f = \bar{\mathbf{A}}_f \bar{\mathbf{x}}_f + \bar{\mathbf{B}}_f \mathbf{u}_f \\ \bar{\mathbf{y}}_f = \bar{\mathbf{C}}_f \bar{\mathbf{x}}_f \end{cases} \quad (3-24)$$

其中输入矩阵 \mathbf{B}_f 变换为如下形式：

$$\bar{\mathbf{B}}_f = \mathbf{T}^{-1} \mathbf{B}_f = \begin{bmatrix} \mathbf{I}_m \\ 0 \end{bmatrix} \quad (3-25)$$

相应的状态矩阵和输出矩阵变换为：

$$\bar{\mathbf{A}}_f = \mathbf{T}^{-1} \mathbf{A}_f \mathbf{T} \quad (3-26)$$

$$\bar{\mathbf{C}}_f = \mathbf{C}_f \mathbf{T} \quad (3-27)$$

由于线性变换不改变闭环系统的特征值，则有：

$$\bar{\mathbf{I}}_{fi} = \mathbf{I}_{fi} = \mathbf{I} \mathbf{i} \quad (3-28)$$

而相应的特征向量变换为：

$$\bar{\mathbf{v}}_i = \mathbf{T}^{-1} \mathbf{v}_i^a \quad (3-29)$$

则有：

$$(\mathbf{I}_i \mathbf{I} - \bar{\mathbf{A}}_f) \bar{\mathbf{v}}_i = \bar{\mathbf{B}}_f \mathbf{K} \bar{\mathbf{C}}_f \bar{\mathbf{v}}_i = \begin{bmatrix} \mathbf{I}_m \\ 0 \end{bmatrix} \mathbf{K} \bar{\mathbf{C}}_f \bar{\mathbf{v}}_i \quad (3-30)$$

为了方便计算，对应于 $\bar{\mathbf{B}}_f$ ，对 $\bar{\mathbf{v}}_i$ 进行相应的分块：

$$\bar{\mathbf{v}}_i = \begin{bmatrix} \bar{\mathbf{z}}_i \\ \bar{\mathbf{w}}_i \end{bmatrix}, \text{ 其中 } \bar{\mathbf{z}}_i \in R^{m \times 1}, \bar{\mathbf{w}}_i \in R^{(n-m) \times 1}$$

同时对 $I_i I - \bar{A}_f$ 进行相应的分块, 则有:

$$\begin{bmatrix} I_i I_m - \bar{A}_{f11} & -\bar{A}_{f12} \\ -\bar{A}_{f21} & I_i I_{n-m} - \bar{A}_{f22} \end{bmatrix} \begin{bmatrix} \bar{z}_i \\ \bar{w}_i \end{bmatrix} = \begin{bmatrix} I_m \\ 0 \end{bmatrix} K \bar{C}_f \begin{bmatrix} \bar{z}_i \\ \bar{w}_i \end{bmatrix} \quad (3-31)$$

由矩阵的第一行矩阵方程可得:

$$I_i I_m \bar{z}_i - \begin{bmatrix} \bar{A}_{f11} & \bar{A}_{f12} \end{bmatrix} \bar{v}_i = K \bar{C}_f \bar{v}_i \quad (3-32)$$

记 $\begin{bmatrix} \bar{A}_{f11} & \bar{A}_{f12} \end{bmatrix}$ 为 \bar{A}_{f1} , 则上式可写为:

$$(\bar{A}_{f1} + K \bar{C}_f) \bar{v}_i = I_i \bar{z}_i \quad (3-33)$$

对于任一期望特征值和对应的可达特征向量, 有下列公式成立:

$$(\bar{A}_{f1} + K \bar{C}_f) \bar{v}_1 = I_1 \bar{z}_1$$

$$(\bar{A}_{f1} + K \bar{C}_f) \bar{v}_2 = I_2 \bar{z}_2$$

....

$$(\bar{A}_{f1} + K \bar{C}_f) \bar{v}_r = I_r \bar{z}_r$$

总的可以写为:

$$(\bar{A}_{f1} + K \bar{C}_f) \bar{V} = \bar{Z} \quad (3-34)$$

其中 $\bar{V} = \begin{bmatrix} \bar{v}_1 & \bar{v}_2 & \mathbf{L} & \bar{v}_r \end{bmatrix}$; $\bar{Z} = \begin{bmatrix} I_1 \bar{z}_1 & I_2 \bar{z}_2 & \mathbf{L} & I_r \bar{z}_r \end{bmatrix}$ 。

特征结构配置所需的反馈增益矩阵 K 可表示为:

$$K = -(\bar{Z} - \bar{A}_{f1} \bar{V})(\bar{C}_f \bar{V})^{-1} \quad (3-35)$$

式中: \bar{A}_{f1} 是矩阵 \bar{A}_f 的分块, $\bar{A}_{f1} = \begin{bmatrix} \bar{A}_{f11} & \bar{A}_{f12} \end{bmatrix}$; \bar{Z} 矩阵的列与 r 个期望特征值及相应的可达特征向量相关。

值得注意的是, 上述重构算法的实现, 是在假定故障不改变系统的可控和可观性基础上的。如果故障后系统的开环特征值与标称系统的闭环特征值有重合元素, 故障系统发生了不可控或者不可观模态, 将导致上面的算法失效。在这种情况下, 采用奇异值分解法的特征结构配置重构可以实现控制器设计^[38,153-161]。

已知对故障系统, 有下式成立:

$$(A_f + B_f K C_f) v_i^f = I_i^f v_i^f \quad (3-36)$$

将上式转换成矩阵方程：

$$\begin{bmatrix} I_i I - A_f \mathbf{M} B_f \\ -K C_f \end{bmatrix} \begin{bmatrix} v_i \\ m_i \end{bmatrix} = \mathbf{0} \quad (3-37)$$

定义 m 维向量 m_i ： $m_i = -K C_f v_i$ ，则式 (3-37) 可写为：

$$\begin{bmatrix} I_i I - A_f \mathbf{M} B_f \\ -K C_f \end{bmatrix} \begin{bmatrix} v_i \\ m_i \end{bmatrix} = \mathbf{0} \quad (3-38)$$

对矩阵 $\begin{bmatrix} I_i I - A_f \mathbf{M} B_f \\ -K C_f \end{bmatrix}$ 进行奇异值分解，则有：

$$\begin{bmatrix} I_i I - A_f \mathbf{M} B_f \\ -K C_f \end{bmatrix} = U S Q^T \quad (3-39)$$

其中 $S \in R^{n \times (n+M)}$ ，且可以写成如下形式：

$$S = \begin{bmatrix} s_i & 0 \\ 0 & 0 \end{bmatrix}, s_i \in R^{n \times r_i} \quad (3-40)$$

因此，式 (3-38) 可转换为：

$$U \begin{bmatrix} s_i & 0 \\ 0 & 0 \end{bmatrix} Q^T \begin{bmatrix} v_i \\ m_i \end{bmatrix} = 0 \quad (3-41)$$

由于 U 矩阵为非奇异矩阵，上式可写为：

$$\begin{bmatrix} s_i & 0 \\ 0 & 0 \end{bmatrix} Q^T \begin{bmatrix} v_i \\ m_i \end{bmatrix} = 0 \quad (3-42)$$

为了方便计算，将 Q^T 写成如下分块矩阵的形式：

$$Q^T = \begin{bmatrix} Q_{i,11} & Q_{i,12} \\ Q_{i,21} & Q_{i,22} \end{bmatrix} \quad (3-43)$$

其中 $Q_{i,11} \in R^{r_i \times n}$ ， $Q_{i,12} \in R^{r_i \times m}$ ， $Q_{i,21} \in R^{(n-r_i) \times n}$ ， $Q_{i,22} \in R^{(n-r_i) \times m}$ ，则式 (3-42) 可转换为：

$$\begin{bmatrix} s_i & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} Q_{i,11} & Q_{i,12} \\ Q_{i,21} & Q_{i,22} \end{bmatrix} \begin{bmatrix} v_i \\ m_i \end{bmatrix} = 0 \quad (3-44)$$

由于 s_i 满秩，有下式成立：

$$\begin{cases} Q_{i,11}v_i^f + Q_{i,12}m_i = 0 \\ Q_{i,21}v_i^f + Q_{i,22}m_i = N_i \end{cases} \quad (3-45)$$

上式中 N_i 是 $n+m-r_i$ 维的任意向量。

求解式 (3-45)，可得：

$$\begin{cases} v_i^f = Q_{i,1}N_i \\ m_i = Q_{i,2}N_i \end{cases} \quad (3-46)$$

其中 $Q_{i,1} \in R^{n \times (n+m-r_i)}$ ， $Q_{i,2} \in R^{m \times (n+m-r_i)}$ ， $Q = \begin{bmatrix} \times & Q_{i,1} \\ \times & Q_{i,2} \end{bmatrix}$ 。

通过选择合适的 N_i 矩阵，可以获得最优可达向量特征向量 v_i^f ，进一步由式 (3-35) 计算反馈增益矩阵 K ，从而使得重构控制器可以最大程度恢复系统的特征结构，满足重构控制设计的性能要求。

3.3 基于改进粒子群优化的特征结构配置重构控制律设计

上文研究的特征结构重构算法可以有效恢复故障后的系统性能，但由于该算法要求故障前后系统维数相等，因此对于故障引起的维数不匹配系统失效；同时由状态空间理论的相关分析可知，基于全状态反馈的特征结构配置可以保证闭环系统的全部 n 个特征值完全恢复，因此保证了重构系统的稳定性，而基于输出反馈的特征结构配置重构方法，只恢复了 $\max(m, p)$ 个主特征值，因此有必要对剩余 $n - \max(m, p)$ 个特征值进行控制，以保证重构的稳定性^[154,157]；本节针对这些情况，结合改进的粒子群寻优算法，提出一种特征结构配置重构控制律设计方案。

对于 (3-7) 式所描述的故障系统方程，采用 (3-8) 式反馈控制形式，可以求解出闭环系统状态时间响应、输出时间响应与系统特征值、特征向量之间的关系：

$$x(t) = e^{(A_f + B_f K C_f)t} x_f(0) + \int_0^t e^{(A_f + B_f K C_f)(t-\tau)} B_f u_f(\tau) d\tau \quad (3-47)$$

$$y(t) = \sum_{i=1}^n (C_f v_i^f) e^{l_i^f} (w_i^f)^T x_f(0) + \sum_{j=1}^m \sum_{i=1}^n (C_f v_i^f) ((w_i^f)^T b_j^f) \int_0^t e^{l_i^f \tau} u_j^f(\tau) d\tau \quad (3-48)$$

上式中 $x_f(0)$ 为故障系统状态初值， w_i^f 和 v_i^f 分别为系统左、右特征向量， b_j^f 为矩阵 B_f 的第 j 列， u_j^f 为矩阵 u_f 的第 j 行，即第 j 个输入量。

可以看到，闭环系统输出时间响应是多个模态的线性叠加， $C_f v_i^f$ ， $(w_i^f)^T x_f(0)$ 和 $(w_i^f)^T b_j^f$ 是各个模态所对应的幅值，决定了模态在时间响应中的关系，这种关系描述了模态与输出的耦合程度，决定了模态相应的大小。因此，为了尽可能恢复故障系统性能，使其响应最大程度的恢复到标称系统，采用特征结构配置重构算法进行控制律设计时，理想特征值可选取标称系统的特征值，理想特征向量的选取可依照下面的最小方差优化函数 J_v ：

$$J_v = \sum_{i=1}^n \left[a_i \|C v_i - C_f v_i^f\|_2 + b_i |w_i^T x(0) - (w_i^f)^T x_f(0)| + g_i \sum_{j=1}^m |w_i^T b_j - (w_i^f)^T b_j^f| \right] \quad (3-49)$$

式中 $x(0)$ 为标称系统状态初值， w_i 和 v_i 分别为系统左、右特征向量， b_j 为矩阵 B_f 的第 j 列，系数 a_i 、 b_i 和 g_i 的选择视具体情况而定，比如说无初始状态时，可令 $b_i = 0$ ；无外部输入时可令 $g_i = 0$ 。

为了保证重构后系统的稳定性，特征结构配置重构设计考虑全部的闭环特征值，对特征值采用最小方差优化函数 J_l ：

$$J_l = \sum_{i=1}^n |l_i^f - l_i| \quad (3-50)$$

对于特征结构配置反馈增益矩阵的求解，考虑到稳定性和不匹配问题，定义综合优化指标函数为：

$$\begin{aligned} J &= w_1 J_v + w_2 J_l \\ &= w_1 \sum_{i=1}^n \left[a_i \|C v_i - C_f v_i^f\|_2 + b_i |w_i^T x(0) - (w_i^f)^T x_f(0)| + g_i \sum_{j=1}^m |w_i^T b_j - (w_i^f)^T b_j^f| \right] \\ &\quad + w_2 \sum_{i=1}^n |l_i^f - l_i| \end{aligned} \quad (3-51)$$

其中， w_1, w_2 为加权系数。

在这个优化指标的基础上，本文结合改进的粒子群算法，将反馈增益矩阵的求解转化为寻优问题。为了提高粒子群的搜索效率，采用特征值灵敏度的思想，公式 (3-50) 可改进为：

$$J_l = \sum_{i=1}^k r_i |l_i^f - l_i| + \sum_{j=k+1}^n h_j |l_j^f - l_j| \quad (3-52)$$

r_i 为主导特征值寻优加权系数， h_j 为非主导特征值寻优加权系数， $r_i \gg h_j$ 。

相应的，公式（3-51）可改进为：

$$\begin{aligned}
 J &= w_1 J_v + w_2 J_v \\
 &= w_1 \sum_{i=1}^n \left[a_i \|Cv_i - C_f v_i^f\|_2 + b_i |w_i^T x(0) - (w_i^f)^T x_f(0)| + g_i \sum_{j=1}^m |w_i^T b_j - (w_i^f)^T b_j^f| \right] \\
 &\quad + w_2 \left[\sum_{i=1}^k r_i |l_i^f - l_i| + \sum_{j=k+1}^n h_j |l_j^f - l_j| \right]
 \end{aligned} \quad (3-53)$$

粒子群优化算法源于对鸟群和鱼群捕食行为的启发，模拟鸟群的捕食行为：假设一群鸟在某个区域随机搜索食物，在这个区域里只有一块食物，所有的鸟都不知道食物在这个区域中的位置，但是它们知道各自当前的位置离区域中食物的距离，那么找到食物最简单有效的方法就是搜寻目前离食物最近的鸟的周围区域。

粒子群优化算法就是从上面描述的模型中得到启示而产生的，是一种基于迭代的群智能优化工具，具有流程简单、算法简洁、全局优化等特点。应用粒子群算法在求解参数优化问题时，问题的解对应于空间 D 中的一个粒子，每个粒子都有自己的位置和速度，在算法实现过程中以粒子对解空间中最优粒子的追随进行解空间的搜索，粒子通过跟踪两个极值来更新自己，一个是粒子本身找到的最优解- $pbest$ ，另一个是整个种群找到的最优解- $gbest$ ，基本粒子群算法中粒子的速度和位置更新方程为：

$$\begin{cases} V_i(t+1) = wV_i(t) \\ \quad + c_1 r_1 \times (pbest_i(t) - X_i(t)) \\ \quad + c_2 r_2 \times (gbest_i(t) - X_i(t)) \\ X_i(t+1) = X_i(t) + V_i(t) \end{cases} \quad (3-54)$$

其中， $V_i(t+1)$ 、 $V_i(t)$ 为粒子 i 在 $t+1$ 和 t 步的速度； $X_i(t+1)$ 、 $X_i(t)$ 为粒子 i 在 $t+1$ 和 t 步的位置。 c_1 和 c_2 称为加速因子， c_1 表示粒子受社会认识的影响程度， c_2 表示粒子受自身经验的影响程度； r_1 和 r_2 为 (0,1) 中的随机数； $pbest_i(t)$ 为粒子 i 在 t 步时的个体极值； $gbest_i(t)$ 为粒子 i 在 t 步时的全局极值； w 为惯性权重，对算法的收敛性有很大的影响，适当的选取 w 的值可以平衡整个粒子群算法的全局探索和局部开发能力，为此本文对惯性权重进行改进，采用指数曲线衰减，其更新式为：

$$w = w_e + (w_s - w_e)e^{-at/T} \quad (3-55)$$

式中, w_s 和 w_e 分别为 w 的初始值和终值; a 系数调节指数曲线的衰减速度; t 为粒子群当前的运算步数; T 为总迭代次数。 w 随着粒子群的运算作指数曲线变化: 在寻优的初始阶段, w 的值比较大, v 较大, 适于对解空间进行大范围的搜索, 即粒子的全局寻优能力强; 随着粒子群迭代运算的进行, w 迅速减少, v 较小, 适于小范围的精细搜索, 可加强粒子的局部开发能力。

由式 (3-54) 可知, 速度向量由三项组成: 第一项为从原有速度继承的速度分量, 代表粒子原有速度对新速度的影响, 具有平衡全局探索和局部开发的能力; 第二项为粒子被个体极值所影响的速度分量, 体现了粒子自身的知识, 使粒子具备足够强的全局探索能力, 即粒子搜索到最优解所在区域的能力; 第三项分为粒子被全局极值所影响的速度分量, 体现了粒子间的信息共享和知识合作, 使粒子具备足够强的局部开发能力, 即粒子在最优解区域找到最优解的能力。在这三部分的作用下, 粒子群算法可以有效的搜索到最优解。

个体极值 $pbest_i(t+1)$ 和的全局极值 $gbest_i(t)$ 更新公式为:

$$pbest_i(t+1) = \begin{cases} pbest_i(t) & \text{if } g(x_i(t+1)) \geq g(pbest_i(t)) \\ x_i(t+1) & \text{if } g(x_i(t+1)) < g(pbest_i(t)) \end{cases} \quad (3-56)$$

$$\begin{aligned} gbest_i(t+1) &= \{pbest_i(t+1) | g(pbest_i(t+1)) \\ &= \min \{g(pbest_0(t+1)), g(pbest_1(t+1)), \mathbf{L}, g(pbest_n(t+1))\} \end{aligned} \quad (3-57)$$

在 (3-56)、(3-57) 中, $g(x)$ 为粒子群的适应度评价函数, 若优化问题的目标函数为 $f(x)$, 则 $g(x)$ 的表达式为:

$$g(x) = \frac{1}{f(x)} \quad (3-58)$$

选取优化目标函数后, 反馈矩阵的元素对应于粒子的位置向量, 粒子初始位置和速度初始值随机产生, 然后按照公式 (3-54) 进行训练, 直到满足优化函数给定的条件为止, 此时得到最优粒子即矩阵的最优值。

基于改进粒子群的特征结构配置反馈矩阵设计如下:

步骤 1, 初始化: 确定粒子群中粒子的维数 D 和群体规模 N , 设定粒子群优化算法的参数, 包括惯性权重的初始值 w_s 和终值 w_e 、加速系数 c_1 和 c_2 、速度的限定值 V_{\max} , 算法的最大迭代次数 T 或迭代终止精度 e , 随机初始化粒子的位置和速度, 将粒子的位置向量解码成对应的寻优参数, 可以得到反馈增益矩阵 K ;

确定综合优化指标函数 J 中的加权系数, 通过反馈增益矩阵 K 可以求解出优化指标函数 J 的初始值;

步骤 2, 将粒子的位置向量解码成对应的寻优参数, 得到粒子群的适应度函数。根据式 (3-56)、(3-57) 计算粒子的个体极值 $pbest$ 和全局极值 $gbest$;

步骤 3, 根据式 (3-54) 对粒子进行速度和位置的更新, 若粒子的速度超出限定值 V_{\max} , 则速度取边界值;

步骤 4, 将粒子的位置向量解码成对应的寻优参数, 检验算法是否满足终止条件, 若满足, 则停止迭代, 输出最优解; 否则转到步骤 2, 进行下一个迭代周期的训练。算法的终止条件为算法达到最大迭代次数 T 或者粒子目标函数达到了给定的迭代终止精度 e , 记录此时得到的最优解即最终优化得到的矩阵。

基于特征结构配置的稳定性重构算法设计流程图如下所示。

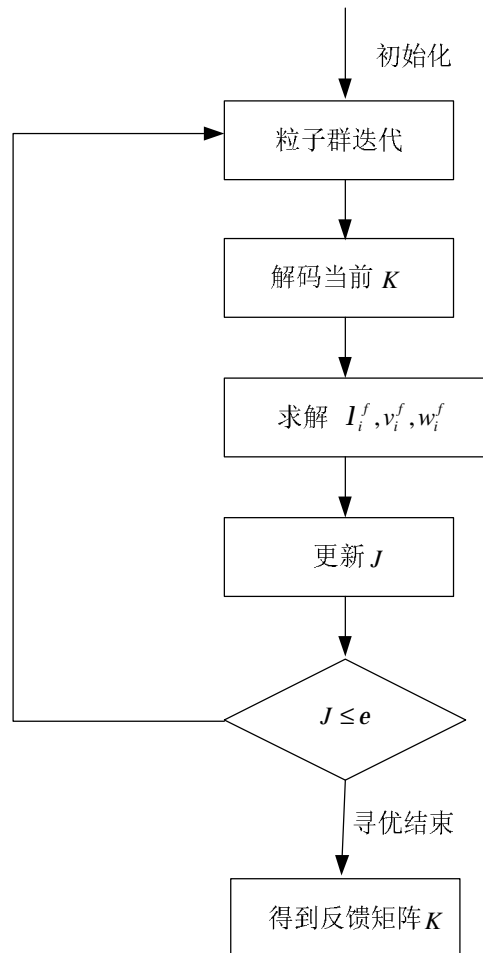


图 3-2 基于改进粒子群的特征结构配置反馈增益矩阵设计流程图

3.4 仿真算例

某型飞机的侧向飞行运动方程描述为（ $H = 5000m, Ma = 0.6$ ）：

$$\begin{cases} \begin{bmatrix} \dot{b} \\ \dot{w}_x \\ \dot{w}_y \\ \dot{g} \end{bmatrix} = \begin{bmatrix} -0.07095 & 0.0962 & 1 & 0.0465 \\ -3.4161 & -0.7046 & -0.1858 & 0 \\ -0.6529 & 0.0287 & -0.1467 & 0 \\ 0 & 1 & -0.0965 & 0 \end{bmatrix} \begin{bmatrix} b \\ w_x \\ w_y \\ g \end{bmatrix} + \begin{bmatrix} 0 & -0.0707 \\ -0.9535 & -1.817 \\ 0.0779 & -29.332 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} d_x \\ d_y \end{bmatrix} \\ \begin{bmatrix} b \\ w_x \\ w_y \\ g \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} b \\ w_x \\ w_y \\ g \end{bmatrix} \end{cases}$$

其中，状态向量 b, w_x, w_y, g 分别表示飞机的侧滑角、滚转速率、偏航速率和滚转角改变量； d_x, d_y 分别为副翼和方向舵偏转角改变量，舵回路均采用一阶惯性环节 $\frac{20}{s+20}$ ，仿真中考虑了舵偏转 $\pm 25^\circ$ 位置饱和限制及舵面偏转速率 $\pm 75^\circ/s$ 饱和限制。

粒子群算法的参数设置为：训练步数 $T = 1000$ ，群体规模 $N = 25$ ，加速因子 $c_1 = c_2 = 2$ ，惯性权重按曲线调整，初始和终止惯性权值为 $w_s = 0.9, w_e = 0.2$ ；

标称状态下，根据飞行品质和输出耦合关系的要求，选取的期望特征值为：

$$I_{1,2} = -2 \pm 2i, I_3 = -2, I_4 = -1;$$

期望特征向量为：

$$v_1 = [1 \ 0 \ 1 \ 0]^T, v_2 = [1 \ 0 \ 1 \ 0]^T, v_3 = [0 \ 1 \ 0 \ 1]^T, v_4 = [0 \ 1 \ 0 \ 1]^T$$

特征结构配置反馈矩阵为：

$$K = \begin{bmatrix} -3.7620 & 2.3840 & 2.0830 \\ 0.2340 & 0.0158 & 0.0049 \end{bmatrix};$$

配置后系统闭环特征值为： $I_{1,2} = -2 \pm 2i, I_3 = -2, I_4 = -1$ ；闭环特征向量为：

$$v_1 = \begin{bmatrix} 0.2331+0.2442i \\ -0.0101 \\ -0.9408 \\ -0.0202-0.0202i \end{bmatrix}, v_2 = \begin{bmatrix} 0.2331-0.2442i \\ -0.0101 \\ -0.9408 \\ -0.0202+0.0202i \end{bmatrix}, v_3 = \begin{bmatrix} -0.0265 \\ 0.8938 \\ -0.0137 \\ -0.4475 \end{bmatrix}, v_4 = \begin{bmatrix} -0.0173 \\ 0.7060 \\ -0.0188 \\ -0.7078 \end{bmatrix}。$$

以参数摄动为例，故障后，系统的输入矩阵为：

$$B_f = \begin{bmatrix} 0 & -0.7 & 0.03 & 0 \\ -0.0707 & -1.817 & -1 & 0 \end{bmatrix}^T，$$

利用本章提出的特征结构重构方法设计新的重构反馈矩阵，特征结构反馈增益矩阵为：

$$K_f = \begin{bmatrix} -21.4725 & 2.4283 & 2.6816 \\ 6.5335 & 0.3311 & 0.0651 \end{bmatrix}。$$

重构后，系统的闭环系统特征值为： $L_{1,2} = -2 \pm 2i, L_3 = -2, L_4 = -1$ ；特征向量为：

$$v_1 = \begin{bmatrix} 0.1781+0.2443i \\ -0.0113+0.0008i \\ -0.9530 \\ -0.0202-0.0204i \end{bmatrix}, v_2 = \begin{bmatrix} 0.1781-0.2443i \\ -0.0113-0.0008i \\ -0.9530 \\ -0.0202+0.0204i \end{bmatrix}, v_3 = \begin{bmatrix} -0.0247 \\ 0.8938 \\ -0.0129 \\ -0.4475 \end{bmatrix}, v_4 = \begin{bmatrix} -0.0165 \\ 0.7060 \\ -0.0179 \\ -0.7078 \end{bmatrix}。$$

将该控制策略应用于侧向协调转弯的飞行过程，协调转弯是指飞机在水平面内连续改变飞行方向，保证 $b=0$ 即滚转与偏航运动两者耦合最小，并保证不掉高度的一种转弯。仿真中，滚转角指令为 $g_d = 30/57.3rad$ ，舵面角位移饱和限制为 $|d|_{\max} = 0.4363rad$ ，舵面角速率饱和限制为 $|\dot{d}|_{\max} = 1.4834rad/s$ 。正常状态和重构控制的仿真结果对比如图 3-3 所示。

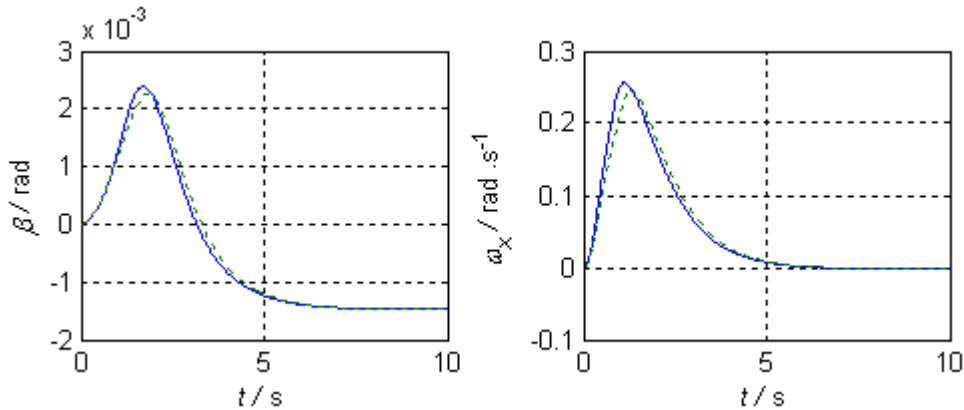


图 3-3-1 参数摄动时侧滑角和滚转角速率输出对比曲线

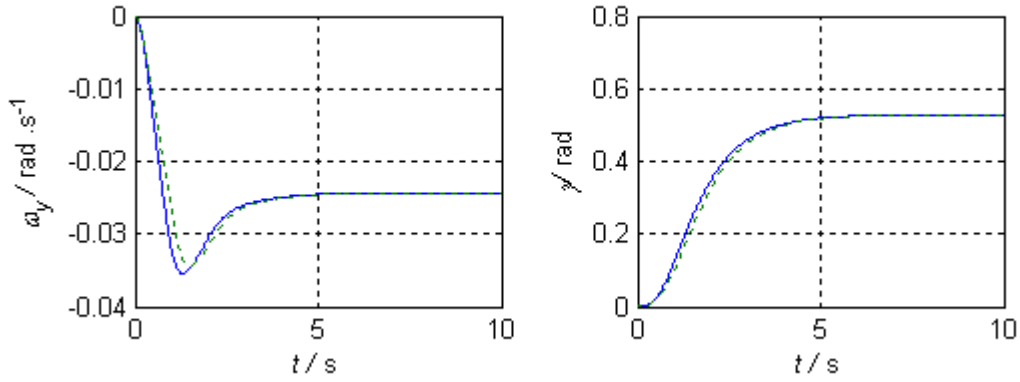


图 3-3-2 参数扰动时偏航角速率和滚转角输出对比曲线

图中虚线表示参数扰动时的重构系统输出，由重构输出与正常情况下输出的对比曲线可以看出，采用特征结构配置重构控制策略的系统可以实现对滚转角的准确跟踪，无稳态误差，调节时间短，在这个过程中侧滑角 β 小，较好的恢复了正常情况下的动态性能和指令跟踪能力。

以副翼损伤 20% 为例，重构后的特征结构反馈增益矩阵为：

$$K_f = \begin{bmatrix} -4.7025 & 2.9800 & 2.6037 \\ 0.2304 & 0.0158 & 0.0049 \end{bmatrix}$$

相应的，系统的闭环系统特征值为： $I_{1,2} = -2 \pm 2i, I_3 = -2, I_4 = -1$ ；特征向量为：

$$v_1 = \begin{bmatrix} 0.9920 \\ -0.0206 + 0.0216i \\ -1.9088 + 1.9999i \\ -0.0837 + 0.0020i \end{bmatrix}, v_2 = \begin{bmatrix} -0.2475 + 0.2593i \\ 0.0107 \\ 0.9990 \\ 0.0214 - 0.0214i \end{bmatrix}, v_3 = \begin{bmatrix} -0.0296 \\ 0.9989 \\ -0.0153 \\ -0.5002 \end{bmatrix}, v_4 = \begin{bmatrix} 0.0245 \\ -0.9961 \\ 0.0265 \\ 0.9987 \end{bmatrix}$$

同样将该控制策略应用于副翼损伤 20% 时的侧向协调转弯飞行过程，滚转角指令为 $g_d = 30/57.3 \text{ rad}$ ，正常状态和重构控制的仿真结果对比如下图所示：

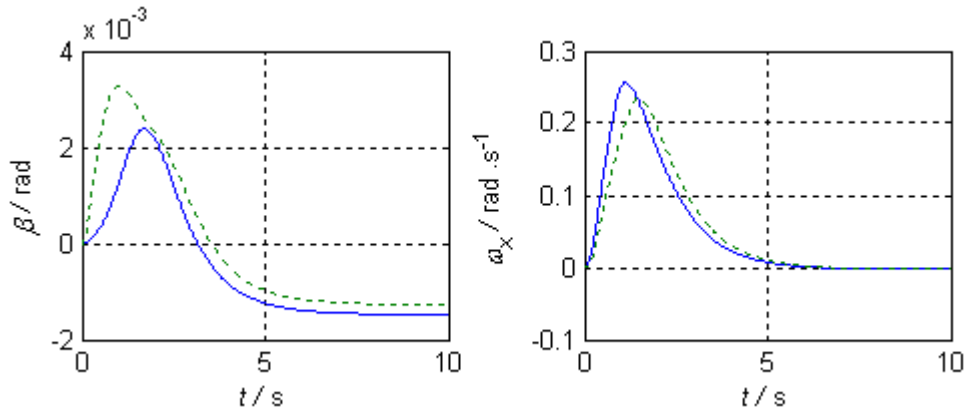


图 3-4-1 副翼损伤 20% 时侧滑角和滚转角速率输出对比曲线

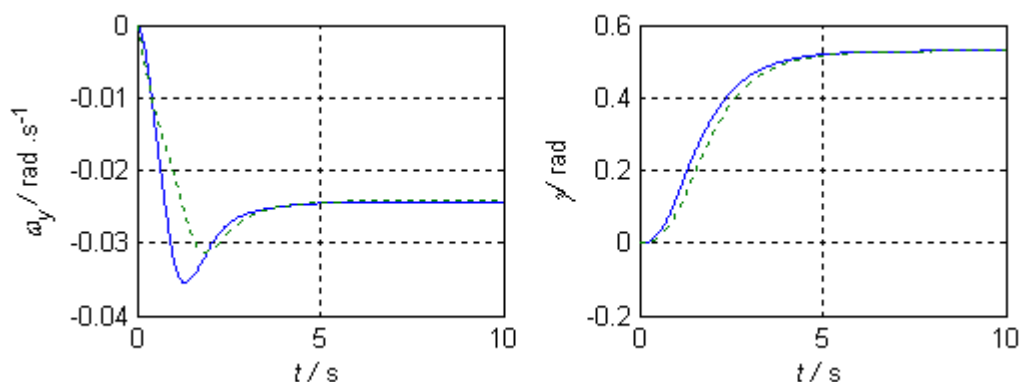


图 3-4 副翼损伤 20%时偏航角速率和滚转角输出对比曲线

可以看出，特征结构配置重构控制系统可以实现对滚转角的准确跟踪，无稳态误差，调节时间短，较好的恢复了正常情况下的动态性能和指令跟踪能力。

当副翼损伤 50%时，采用特征结构配置重构的系统响应曲线如下图所示：

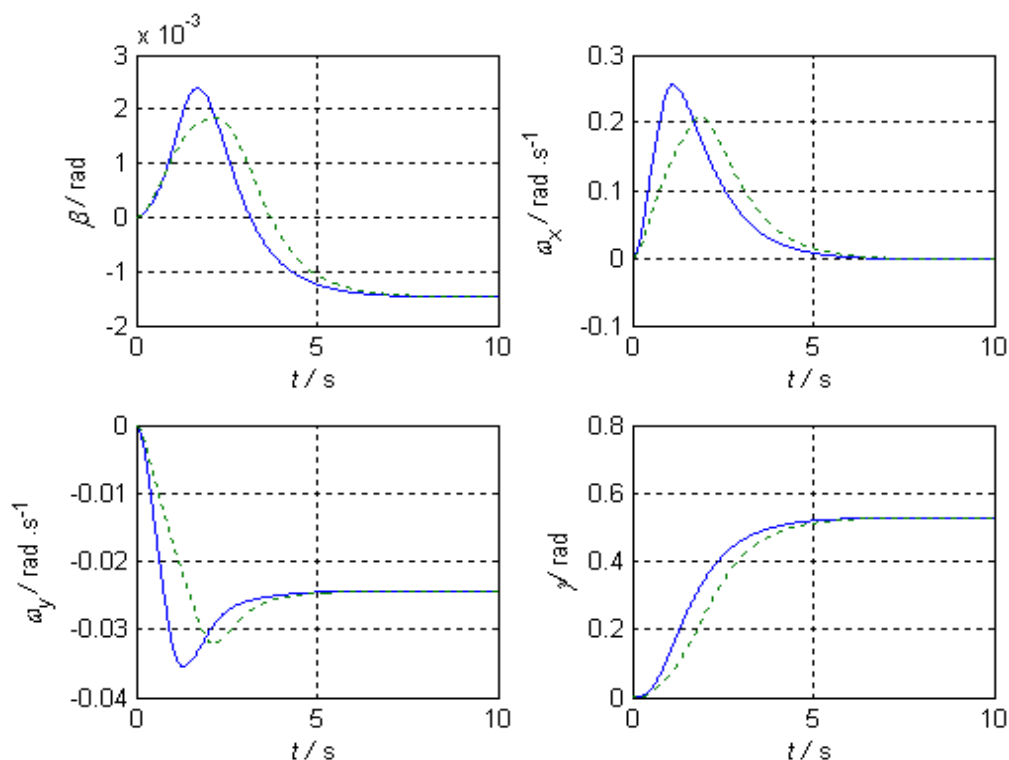


图 3-5 副翼损伤 50%系统输出曲线

当方向舵损伤 50%时，采用特征结构配置重构的系统响应曲线如下图所示：

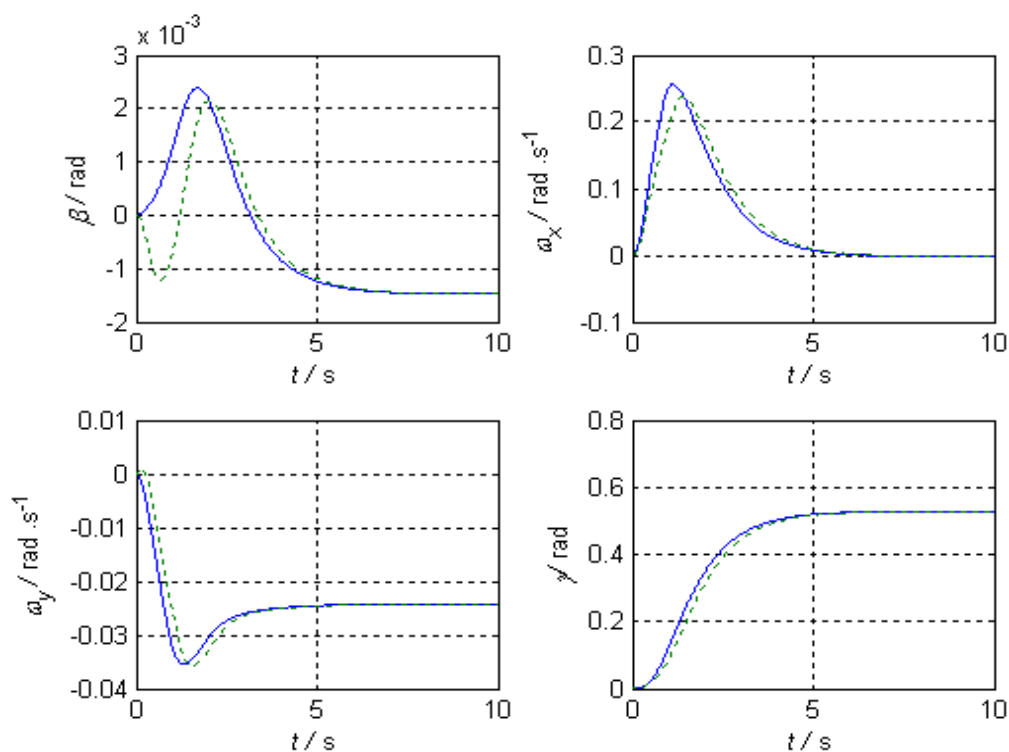


图 3-6 方向舵损伤 50% 系统输出曲线

当方向舵损伤松浮时，故障后系统和原系统维数不匹配，采用特征结构配置重构的系统响应曲线如下图所示：

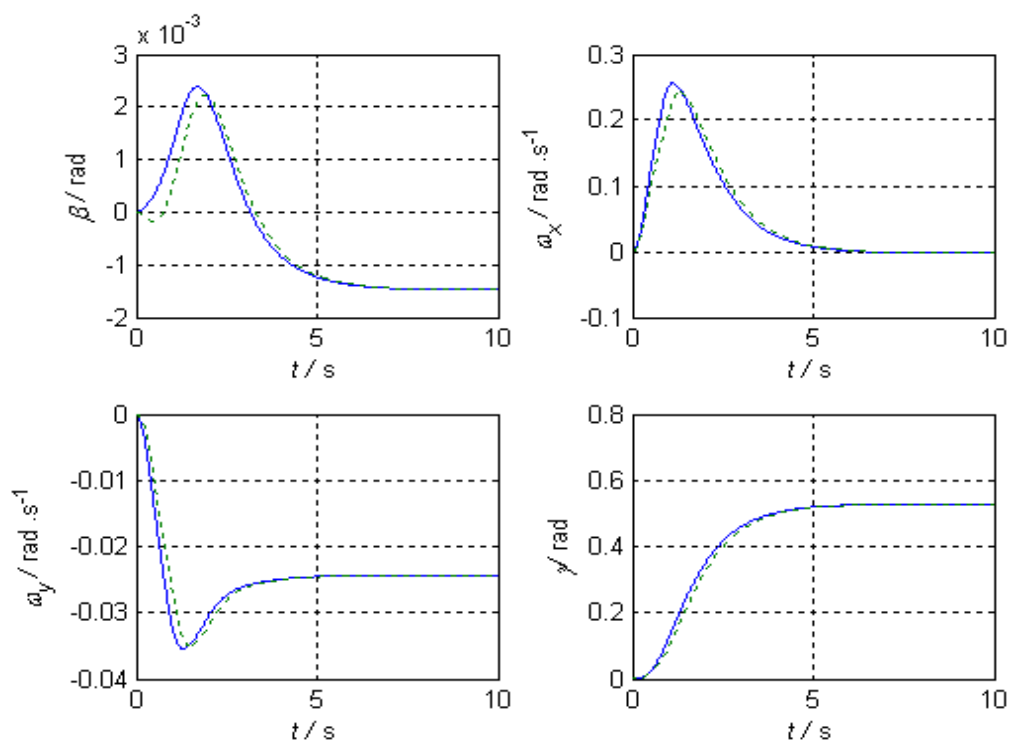


图 3-7 方向舵松浮系统输出曲线

可见采用特征结构结构配置的重构控制策略通过恢复系统的特征值和特征

向量，保证了系统的动态性能和稳定性。将该控制策略应用于容错飞行控制中，重构飞行系统恢复了较好的飞行品质。

3.5 小结

本章采用基于输出反馈的特征结构配置方法设计了容错飞行控制律，该方法通过设计特征结构配置反馈增益矩阵，使得故障系统的特征值和特征向量得到最优恢复，从而保证重构系统的性能最大程度恢复到正常系统的性能。本章考虑到故障可能引起的维数不匹配情况，以及传统的基于输出反馈的特征结构配置重构方法仅能配置 $\max(m, p)$ 个特征值而可能引起的系统不稳定性，提出一种综合了特征值灵敏度和特征向量的寻优设计指标，在此基础上结合改进的粒子群算法，提出一种特征结构配置重构控制设计方法。

第四章 基于简单自适应的容错飞行控制设计

飞行控制系统的高可靠性是飞行设计的重要要求,为了保证飞行任务的顺利完成,有必要对故障系统进行容错控制设计。上文讨论的基于特征结构配置的重构设计方法要求知道系统的精确故障模型,根据故障信息设计相应的重构控制律以保证飞行品质。而基于模型参考自适应方法的重构控制律设计不需要了解具体的故障信息,只关心理想的飞行品质要求,对故障系统进行容错控制律设计,使得故障系统的输出跟踪参考模型的输出以达到设计要求。本章在模型参考自适应理论的基础上,进一步研究了简单自适应控制及该方法在容错控制设计中的应用。

4.1 引言

模型参考自适应控制 (Model-Reference Adaptive Control, MRAC) 是目前很有生命力的一个研究方向,该方法的基本原理是:根据控制要求,在系统中设定一个理想的参考模型,将参考模型输出和被控对象输出之差(称为广义误差)输入到自适应机构来调整控制器参数,以使得误差趋向于零,最终实现被控对象的输出特性跟踪参考模型输出,达到设计要求。其中,并联模型参考自适应是常用的一种参考方案,基本结构如下图所示:

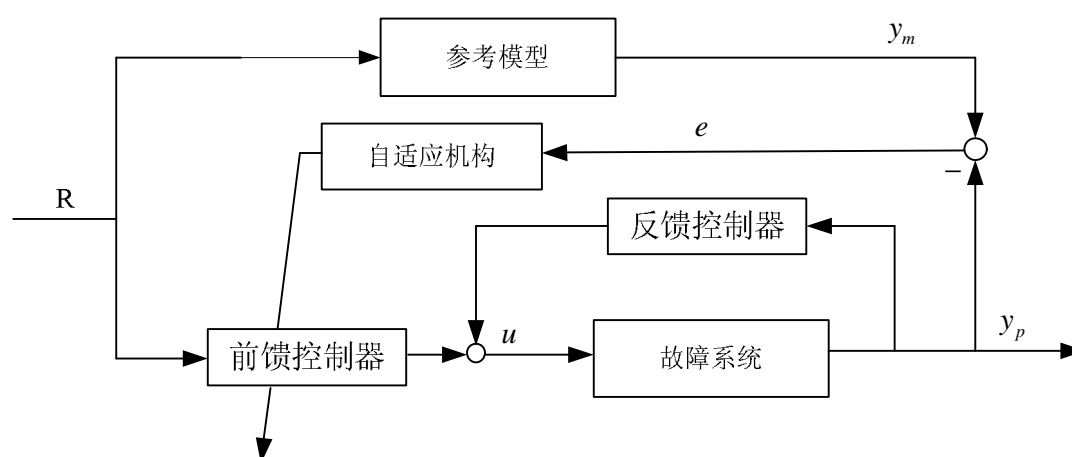


图 4-1 模型参考自适应控制系统结构图

上图中, R 为参考输入, y_m 为模型输出, y_p 为被控对象输出, e 为控制误差, u 为被控对象的控制输入。前馈控制器、被控对象以及反馈控制器构成了可

调系统。在这种自适应控制方案中，被控对象的性能与理想参考模型直接比较，自适应速度较快，工程上也较易实现。

理想的模型参考自适应控制系统设计需要做这样的基本假设：(1) 参考模型为线性定常系统；(2) 可调系统的维数与参考模型的维数一致；(3) 可调系统的参数调整仅依赖于自适应机构；(4) 除输入量之外，无其它外部信号或干扰作用于系统；(5) 广义误差 e 为可测的。

根据控制器参数更新方法的不同，模型参考自适应控制可分为简单自适应控制（直接自适应控制）和间接自适应控制。简单自适应控制于 1979 年由 H.Kaufman, K.Sobel 等学者提出^[30]，由于算法简单、设计方便，该方法在工程中得到了广泛应用。简单自适应控制作为自适应控制的分支，与常规的模型参考自适应控制的设计目标相同，都是根据性能品质，选定理想的参考模型，通过自适应算法使得被控系统的输出特性与参考模型的输出特性保持一致；与常规模型参考自适应理论不同的是，简单自适应控制是建立在指令跟踪发生器或输出渐进跟踪理论上的自适应控制方法，这种算法充分调动了参考模型的信息，利用参考模型的输入信息、输出信息以及系统输出和参考模型输出之间的误差来调节自适应控制器参数，使得系统自动的跟随参考模型的输出。在自适应控制器设计中，在满足系统稳定性条件的前提下，只需给出参考增益矩阵的初值，就可以使得自适应机制自动完成控制器的设计。

相比常规模型跟踪自适应解决问题的方法和过程，简单自适应方法不仅具有控制结构简单，可调参数少；控制系统的设计几乎与被控对象无关，控制器的设计不依赖于被控对象，只是利用参考模型和跟踪误差信息完成设计，实现系统对参考模型的渐进跟踪等优点，更具有适用于单变量系统和多变量系统；可实现高阶被控对象对低阶理想参考模型的性能跟踪；可以构成通用的自适应控制器应用于不同的被控对象等诸多优点，因此目前该方法已经被广泛应用于飞行控制系统设计中。

H.Kaufman 等学者最初提出简单自适应控制方法，并进行了深入的探讨；Itzhak Barkana 和 Kenneth Sobel 进一步研究将这种方法扩展到时变和非线性的被控对象上，并且分析论证了系统的鲁棒性和稳定性，使得简单自适应控制方法不断完善和成熟^[31-36]。

本章采用简单自适应控制方法对飞行控制系统进行了容错设计。文中首先详细论述了简单自适应控制方法的理论基础——指令跟踪发生器理论，由此推导出简单自适应的基本算法，并进一步引入误差积分改进了该算法的鲁棒性；考虑到传统 PI 型调整律的不足，为了使控制律设计更加简单，本章采用二次型性能指标，提出一种基于变步长修正的简单自适应容错控制设计方法；为了保证系统的

稳定性，简单自适应方法要求系统满足 ASPR 条件，本文分别设计了并联前馈补偿器和反馈补偿器，从而保证了被控对象的严格正实性以满足容错控制的需求；最后利用文中提出的控制策略对某型飞机的飞行控制系统进行容错控制，仿真结果表明，采用简单自适应容错控制的飞机在故障情况下仍能保持较为理想的输出特性。

4.2 简单自适应容错控制

4.2.1 问题描述

对于正常情况下的线性多变量连续系统，其状态方程描述为：

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx \end{cases} \quad (4-1)$$

式中， $x \in R^n$ 为系统状态变量， $u \in R^m$ 为系统输入， $y \in R^p$ 为系统输出， $A \in R^{n \times n}$ 为系统矩阵， $B \in R^{n \times m}$ 为系统输入矩阵， $C \in R^{p \times n}$ 为系统输出矩阵。

故障后，系统参数发生变化，状态方程相应将会发生改变。当发生未知故障时，即并不知道故障发生的时间和具体故障参数信息，选择理想的参考模型为：

$$\begin{cases} \dot{x}_m = A_m x_m + B_m u_m \\ y_m = C_m x_m \end{cases} \quad (4-2)$$

其中， $x_m \in R^{n_m}$ 为参考模型状态变量， $u_m \in R^{m_m}$ 为参考模型输入， $y_m \in R^{p_m}$ 为参考模型输出， $A_m \in R^{n_m \times n_m}$ 为参考模型系统矩阵， $B_m \in R^{n_m \times m_m}$ 为参考模型输入矩阵， $C_m \in R^{p_m \times n_m}$ 为参考模型输出矩阵。

容错控制的目标为当被控对象参数未知，即发生未知故障时，设计容错控制器以保证故障后的系统稳定，并且故障系统的输出可以渐进跟踪理想参考模型输出 y_m 。设计的难点在于预先不知道故障发生的具体时间以及故障参数的大小，因此控制器必须具有较强的自适应能力。

4.2.2 基于指令跟踪发生器理论的简单自适应控制

简单自适应控制做为自适应控制理论为一个分支，与常规的模型参考自适应

控制理论的出发点不同，它是基于指令跟踪发生器理论或输出渐进跟踪理论的新型自适应控制方法。指令跟踪发生器理论于 20 世纪 80 年代提出，该理论的基本思想是利用前馈环节以及跟踪误差信息对系统设计理想的控制输入，从而实现被控对象的输出跟踪参考模型输出^[36,37]。

首先对于故障系统和理想参考模型，定义跟踪误差为：

$$e(t) = y_m(t) - y(t) \quad (4-3)$$

存在理想的控制输入 $u^*(t)$ ，可以使得此时的系统输出 $y^*(t)$ 跟踪参考模型的输出，即：

$$e^*(t) = y_m(t) - y^*(t) = 0 \quad (4-4)$$

相应的，此时故障系统的状态向量表示为 $x^*(t)$ ，系统状态方程满足：

$$\begin{cases} \dot{x}^*(t) = A_f x^*(t) + B_f u^*(t) \\ y^*(t) = C_f x^*(t) \end{cases} \quad (4-5)$$

根据跟踪控制理论，系统理想状态时的轨迹应为参考模型状态和参考模型输入的线性组合，即有下式成立：

$$\begin{bmatrix} x^*(t) \\ u^*(t) \end{bmatrix} = \begin{bmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{bmatrix} \begin{bmatrix} x_m \\ u_m \end{bmatrix} \quad (4-6)$$

可知，

$$\dot{x}^*(t) = s_{11} \dot{x}_m(t) + s_{12} \dot{u}_m(t) \quad (4-7)$$

且由于 $\dot{u}_m(t) = 0$ ，则有：

$$\begin{cases} \dot{x}^*(t) = s_{11} A_m x_m(t) + s_{11} B_m u_m(t) \\ y^*(t) = C_m x_m(t) \end{cases} \quad (4-8)$$

由式 (4-5) 和式 (4-6) 可知：

$$\begin{bmatrix} \dot{x}^*(t) \\ y^*(t) \end{bmatrix} = \begin{bmatrix} A_f & B_f \\ C_f & 0 \end{bmatrix} \begin{bmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{bmatrix} \begin{bmatrix} x_m \\ u_m \end{bmatrix} \quad (4-9)$$

由 (4-8) 和 (4-9) 可知：

$$\begin{bmatrix} \dot{x}^*(t) \\ y^*(t) \end{bmatrix} = \begin{bmatrix} s_{11} A_m & s_{11} B_m \\ C_m & 0 \end{bmatrix} \begin{bmatrix} x_m \\ u_m \end{bmatrix} = \begin{bmatrix} A_f & B_f \\ C_f & 0 \end{bmatrix} \begin{bmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{bmatrix} \begin{bmatrix} x_m \\ u_m \end{bmatrix} \quad (4-10)$$

即有下式成立：

$$\begin{bmatrix} s_{11}A_m & s_{11}B_m \\ C_m & 0 \end{bmatrix} = \begin{bmatrix} A_f & B_f \\ C_f & 0 \end{bmatrix} \begin{bmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{bmatrix} \quad (4-11)$$

式 (4-11) 可写为:

$$\begin{cases} s_{11}A_m = A_f s_{11} + B_f s_{21} \\ s_{11}B_m = A_f s_{12} + B_f s_{22} \\ C_m = C_f s_{11} \\ C_f s_{12} = 0 \end{cases} \quad (4-12)$$

若被控对象和参考模型满足

$$\text{rank} \begin{bmatrix} A_f & B_f \\ C_f & 0 \end{bmatrix} = n + m \quad (4-13)$$

可定义:

$$\begin{bmatrix} A_f & B_f \\ C_f & 0 \end{bmatrix} \begin{bmatrix} \Omega_1 & \Omega_2 \\ \Omega_3 & \Omega_4 \end{bmatrix} = I \quad (4-14)$$

值得注意的是: 式 (4-13) 成立要求 $m = p$, 即系统输入量的个数等于输出量的

个数; 如果 $m > p$, 可通过求解伪逆的方法求得矩阵 $\begin{bmatrix} \Omega_1 & \Omega_2 \\ \Omega_3 & \Omega_4 \end{bmatrix}$; 如果 $m < p$, 方

程无解, 基于指令跟踪发生器理论的控制方法失效。

对于式 (4-12), 可得到如下解:

$$\begin{cases} s_{11} = \Omega_1 s_{11} A_m + \Omega_2 C_m \\ s_{12} = \Omega_1 s_{11} B_m \\ s_{21} = \Omega_2 s_{11} A_m + \Omega_3 C_m \\ s_{22} = \Omega_2 s_{11} B_m \end{cases} \quad (4-15)$$

基于指令跟踪发生器理论的简单自适应算法, 其本质就是将参考模型作为一个信号发生器, 自适应增益 k_x 和 k_u 实际是由 s_{21} 和 s_{22} 推广来的。

同样, 首先对于故障系统和理想参考模型, 控制输入为:

$$u(t) = k_e(t)(y_m - y) + k_x(t)x_m + k_u(t)u_m \quad (4-16)$$

其中, $k_x(t)$ 和 $k_u(t)$ 分别为参考模型状态和参考模型输入的前馈增益, k_e 为跟踪误差增益。

$$\text{跟踪误差为:} \quad e(t) = y_m(t) - y(t) \quad (4-17)$$

使用理想的控制输入 $u^*(t)$ 可以使得系统输出跟踪参考模型的输出, 即:

$$e^*(t) = y_m(t) - y^*(t) = 0 \quad (4-18)$$

此时故障系统的状态方程满足：

$$\begin{cases} \dot{x}^*(t) = A_f x^*(t) + B_f u^*(t) \\ y^*(t) = C_f x^*(t) \end{cases} \quad (4-19)$$

且有：

$$u^*(t) = k_x(t)x_m + k_u(t)u_m \quad (4-20)$$

由于系统理想状态时的轨迹应为参考模型状态和参考模型输入的线性组合，有下式成立：

$$\begin{bmatrix} x^*(t) \\ u^*(t) \end{bmatrix} = \begin{bmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{bmatrix} \begin{bmatrix} x_m \\ u_m \end{bmatrix} = \begin{bmatrix} s_{11} & s_{12} \\ k_x & k_u \end{bmatrix} \begin{bmatrix} x_m \\ u_m \end{bmatrix} \quad (4-21)$$

则有：

$$x^*(t) = s_{11}x_m(t) + s_{12}u_m(t) \quad (4-22)$$

且由于 $\dot{u}_m(t) = 0$ ，可知：

$$\dot{x}^*(t) = s_{11}A_m x_m(t) + s_{11}B_m u_m(t) \quad (4-23)$$

由式（4-2），（4-17）（4-19）、（4-21）和（4-23）知，有下面的矩阵方程成立：

$$\begin{bmatrix} A_f & B_f \\ C_f & 0 \end{bmatrix} \begin{bmatrix} s_{11} & s_{12} \\ k_x & k_u \end{bmatrix} \begin{bmatrix} x_m \\ u_m \end{bmatrix} = \begin{bmatrix} s_{11}A_m & s_{11}B_m \\ C_m & 0 \end{bmatrix} \begin{bmatrix} x_m \\ u_m \end{bmatrix} \quad (4-24)$$

若被控对象和参考模型满足

$$\text{rank} \begin{bmatrix} A_f & B_f \\ C_f & 0 \end{bmatrix} = n + m \quad (4-25)$$

定义：

$$\begin{bmatrix} A_f & B_f \\ C_f & 0 \end{bmatrix} \begin{bmatrix} \Omega_1 & \Omega_2 \\ \Omega_3 & \Omega_4 \end{bmatrix} = I \quad (4-26)$$

对于式（4-24），可得到如下解：

$$\begin{cases} s_{11} = \Omega_1 s_{11} A_m + \Omega_2 C_m \\ s_{12} = \Omega_1 s_{11} B_m \\ k_x = \Omega_2 s_{11} A_m + \Omega_2 C_m \\ k_u = \Omega_2 s_{11} B_m \end{cases} \quad (4-27)$$

4.2.3 简单自适应鲁棒容错控制设计

基于指令跟踪发生器理论的简单自适应控制可以在被控对象发生故障、存在未知参数的情况下，对故障系统进行容错控制设计。在保证闭环系统稳定的前提下，采用简单自适应方法设计控制器参数代替常值增益控制器，可以实现被控对象输出跟踪参考模型输出，满足容错设计要求^[36,37,109]。

简单自适应控制的基本结构图如下：

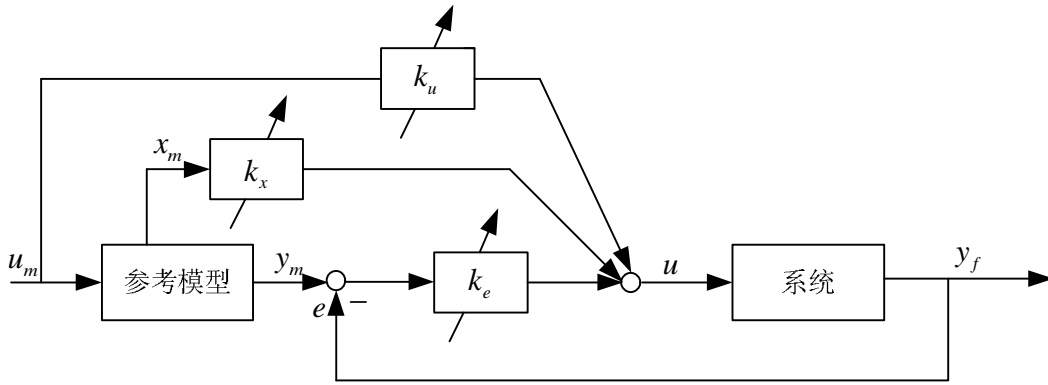


图 4-2 简单自适应控制系统结构图

简单自适应控制输入为：

$$u(t) = k_e(t)e_y(t) + k_x(t)x_m(t) + k_u(t)u_m(t) = K(t)r(t) \quad (4-28)$$

上式中，

$$e_y(t) = y_m(t) - y(t) \quad (4-29)$$

$$K(t) = [k_e(t) \quad k_x(t) \quad k_u(t)] \quad (4-30)$$

$$r^T(t) = [e_y^T(t) \quad x_m^T(t) \quad u_m^T(t)] \quad (4-31)$$

常规的自适应增益矩阵 $K(t)$ 采用 PI 型控制律在线调整，其公式如下：

$$K(t) = K_p(t) + K_i(t) \quad (4-32)$$

$$K_p(t) = e_y(t)r^T(t)T_p, T_p = T_p^T > 0 \quad (4-33)$$

$$\dot{K}_i(t) = e_y(t)r^T(t)T_i, T_i = T_i^T > 0 \quad (4-34)$$

其中， $K_p(t)$ 为比例增益， $K_i(t)$ 为积分增益， T_p, T_i 为常值权矩阵。

在实际飞行过程中，考虑到外部扰动的存在，为了增加简单自适应算法的鲁

棒性，本文引进误差积分抑制外部扰动，自适应增益矩阵 $K(t)$ 的调整公式改进为如下形式：

$$K(t) = K_p(t) + K_i(t) \quad (4-35)$$

$$K_p(t) = e_y(t)r^T(t)T_p, T_p = T_p^T > 0 \quad (4-36)$$

$$\dot{K}_i(t) = e_y(t)r^T(t)T_i + sK_i(t), T_i = T_i^T > 0, s < 0 \quad (4-37)$$

鲁棒性改进算法并没有改变简单自适应的算法结构，只是积分增益调整公式（4-37）与基本算法（4-34）不同，采用改进的 PI 型控制律可以在存在外部扰动的情况下保持自适应系统的鲁棒性。

为了在保持算法鲁棒性的同时，使得跟踪误差尽可能的小，保证输出理想跟踪，在式（4-37）中， s 通常不定义为常数，而是选择可变参数，且 s 随着跟踪误差的减小而减小，其调整公式可采用：

$$s(t) = \frac{s_0 e_y^T(t)e_y(t)}{e_y^T(t)e_y(t) + I}, s_0 < 0 \quad (4-38)$$

选择合适的 T_p ， T_i 和 s_0 ，就可以完成简单自适应算法控制器部分的设计。

4.2.4 基于变步长修正的简单自适应容错控制设计

在上面的研究中，自适应增益矩阵 $K(t)$ 都是采用经典的 PI 型控制律在线调整，这种方法通常对被控对象要求过严，且控制律计算比较难。为了改善 PI 型调整律的不足，使控制律设计更加简单，本节采用二次型性能指标，提出一种变步长修正的简单自适应容错控制设计方法，通过使得二次型性能指标最小来在线设计自适应增益矩阵。

对于式（4-2）描述的理想参考模型，基于变步长修正的简单自适应控制输入为：

$$u(t) = k_e(t)e_y(t) + k_x(t)x_m(t) + k_u(t)u_m(t) = K(t)r(t) \quad (4-39)$$

上式中，

$$e_y(t) = y_m(t) - y(t) \quad (4-40)$$

$$K(t) = [k_e(t) \quad k_x(t) \quad k_u(t)] \quad (4-41)$$

$$r^T(t) = [e_y^T(t) \quad x_m^T(t) \quad u_m^T(t)] \quad (4-42)$$

控制目标设定为：存在理想的控制输入 $u^*(t)$ ，可以使得此时的系统输出 $y^*(t)$ 跟踪参考模型的输出，即：

$$e_y^*(t) = y_m(t) - y^*(t) = 0$$

因此二次型性能指标定义为：

$$J = e_y^T(t) P e_y(t) \quad (4-43)$$

其中， P 为加权矩阵，是一个正实数对角线矩阵。

本文提出一种变步长修正策略作为自适应增益矩阵 $K(t)$ 的调整算法如下：

$$\dot{K}(t) = -w \frac{\partial J}{\partial K} \quad (4-44)$$

其中 w 为修正速度因子，采用指数曲线衰减，其更新式为：

$$w = w_e + (w_s - w_e)e^{-at} \quad (4-45)$$

式中， w_s 和 w_e 为常数， a 系数调节指数曲线的衰减速度。采用这种调整策略，在自适应调整的初始阶段， w 的值比较大，调整步长较大，适于对自适应增益矩阵进行较为快速的调整；随着调整的进行、误差的减小， w 也减少，调整步长较小，适于小范围的精细修正，减小调节误差，尽可能避免震荡，利于得到最优的控制器。其基本结构如下图所示：

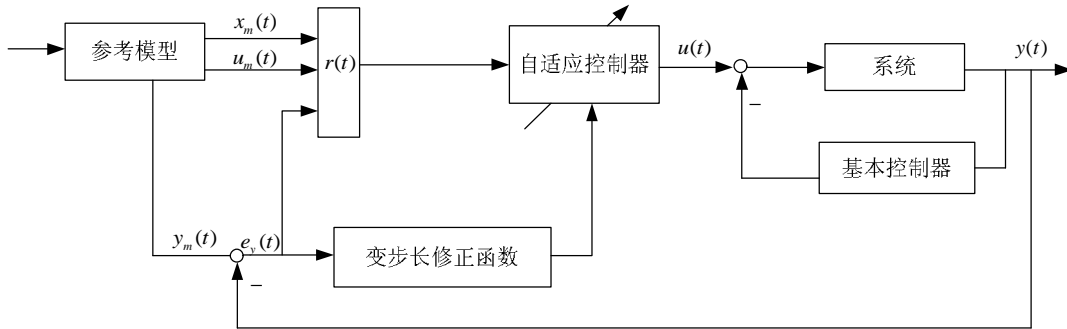


图 4-3 基于变步长修正的简单自适应控制系统结构图

基于变步长修正的设计方法保持了简单自适应控制算法结构简单的特点，与传统的 PI 型设计方法相比，这种方案控制律实现容易，适用于具有一定非线性的被控对象，并能适应被控对象的环境变化。

除了采用上述的变步长修正自适应增益的方法，自适应调节器还可以采用其他算法实现，随着自适应控制理论的扩展，简单自适应算法的实现形式也会更加丰富。

4.3 简单自适应方法的正实性条件

前文提到简单自适应控制要求满足闭环系统稳定的条件才能保证跟踪误差渐进为零，实现完全跟踪。因此，采用简单自适应方法的前提条件是要求系统满足正实性。正实性概念和超稳定性理论是自适应控制稳定性的理论基础，在自适应控制中起着重要的作用，下面引入正实性概念及超稳定性理论的基本概念^[108]。

正实性概念最早是在电工网络分析与综合中提出，表现出网络中能量的非负性，其相应传递函数是正实的，进而推广到最优控制和自适应控制等领域，系统传递函数的正实性概念在自适应控制中有着重要的意义。在自适应控制中，卡尔曼-雅库鲍维奇-波波夫正实性引理是一个很重要的理论，可阐述如下：

系统方程描述为：

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Ju \end{cases} \quad (4-46)$$

其中 A, B 完全可控； A, C 完全可测。传递函数矩阵 $H(s) = J + C[sI - A]^{-1}B$ 为正实函数矩阵的充要条件是存在实矩阵 K, L 和实正定对称矩阵 P ，保证下列 3 个方程式成立。

$$PA + A^T P = -LL^T \quad (4-47)$$

$$B^T P + K^T L^T = C \quad (4-48)$$

$$K^T K = J + J^T \quad (4-49)$$

进一步，若式 (4-47) 换为：

$$PA + A^T P = -LL^T = -Q \quad (4-50)$$

上式中 $Q = Q^T > 0$ ，则传递函数矩阵 $H(s) = J + C[sI - A]^{-1}B$ 为严格正实函数矩阵。

超稳定性理论：在一个闭环系统中，其前向环节为 $G(s)$ ，反馈环节满足波波夫积分不等式。当传递函数矩阵 $G(s)$ 为正实时，这个闭环系统为超稳定系统；当 $G(s)$ 为严格正实时，这个闭环系统为渐近超稳定性系统。

为了保证简单自适应控制算法的收敛性和控制系统的稳定性，我们要求被控对象满足几乎严格正实性条件 (ASPR)，ASPR 条件比较苛刻，很多实际系统并

不满足，对此国内外学者做了相应的研究工作，提出以增加前馈环节或者反馈环节构建扩展系统的方法满足 ASPR^[44,45]。本文在此基础上结合优化算法，进一步提高了算法的质量，保证了容错控制的需求。

4.3.1 应用前馈补偿器保证正实性条件

为了使被控对象满足正实性条件，Bar-Kana 等人^[46]采用并联前馈补偿器构建了扩展系统，其基本结构如下图：

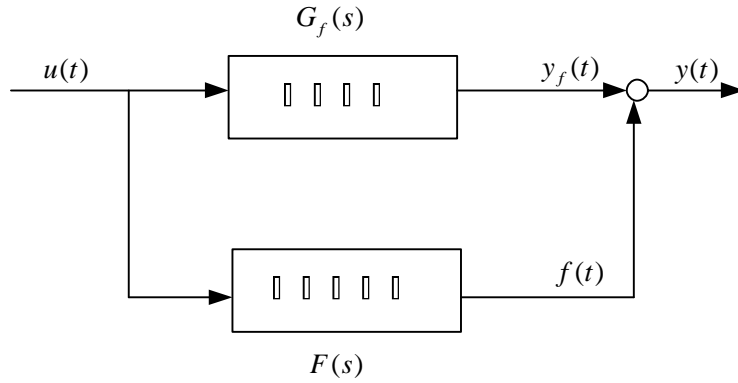


图 4-3 采用前馈补偿器的扩展系统

扩展系统描述为：

$$G(s) = \frac{y(s)}{u(s)} = G_f(s) + F(s) \quad (4-51)$$

对于 $F(s)$ ，有如下结论：

假设 $H(s)$ 为 k 阶反馈传递函数矩阵， $k \leq n$ ，可以使得闭环传递函数 $G_{cl}(s) = [I + G_f(s)H(s)]^{-1}G_f(s)$ 渐近稳定，那么开环系统 $G_o(s) = G_f(s) + H^{-1}(s)$ 满足正实性。由此可知，取前馈补偿器 $F(s) = H^{-1}(s)$ ，可使得扩展系统 $G(s) = G_f(s) + F(s) = G_f(s) + H^{-1}(s)$ 满足正实性。例如前馈补偿器的形式为：

$F(s) = H^{-1}(s) = \frac{K}{1+ts}$ ，只要 $H(s)$ 使得闭环传递函数 $G_{cl}(s) = [I + G_f(s)H(s)]^{-1}G_f(s)$ 渐近稳定，那么开环系统 $G(s) = G_f(s) + H^{-1}(s) = G_f(s) + \frac{K}{1+ts}$ 满足正实性。

值得注意的是，我们希望 $F(s)$ 取值越小越好，当 $F(s)$ 取值很小时，

$G(s) = G_f(s) + F(s) \approx G_f(s)$ ，在这样的情况下，对扩展系统采用简单自适应控制，得到理想输出 $y(t)$ 跟踪 $y_m(t)$ ，且有 $y(t) = y_f(t) + f(t) \approx y_f(t)$ 。

根据上面的讨论，结合优化算法，并充分考虑系统的建模误差，以飞行系统作动器故障为例，考虑作动器 50% 以内的损伤设计鲁棒前馈补偿器，方案如下：

步骤 1：计算当前的 $G_f(s)$ ，前馈补偿器的形式为： $F(s) = H^{-1}(s) = \frac{K}{1+ts}$ ，

求解保证闭环传递函数 $G_{cl}(s) = [I + G_f(s)H(s)]^{-1}G_f(s)$ 渐近稳定的 K 。即利用一个保证闭环系统稳定的反馈器 $H(s)$ 的逆（一个简单的低通滤波器）作为前馈补偿器；

步骤 2：重复上面的步骤，对于所有 B_f ，求解满足正实性条件的 K 交集；

步骤 3：采用优化指标 $J = \min \|K\|$ ，选择最优前馈补偿器 $\frac{K}{1+ts}$ ；

4.3.2 应用反馈补偿器保证正实性条件

对于本文研究的飞行系统，卡尔曼-雅库鲍维奇-波波夫正实性引理可描述如下。

故障系统方程描述为：

$$\begin{cases} \dot{x}_f = A_f x + B_f u \\ y_f = C_f x \end{cases} \quad (4-52)$$

其中 A_f, B_f 完全可控； A_f, C_f 完全可测。传递函数矩阵 $H(s) = C_f[sI - A_f]^{-1}B_f$ 为正实函数矩阵的充要条件是存在实正定对称矩阵 P 和半正定对称矩阵 Q ，保证下列 2 个方程式成立：

$$PA_f + A_f^T P = -Q \quad (4-53)$$

$$B_f^T P = C_f \quad (4-54)$$

进一步，我们可以得到如下引理：

对于式（4-52），采用反馈补偿器构建扩展系统：

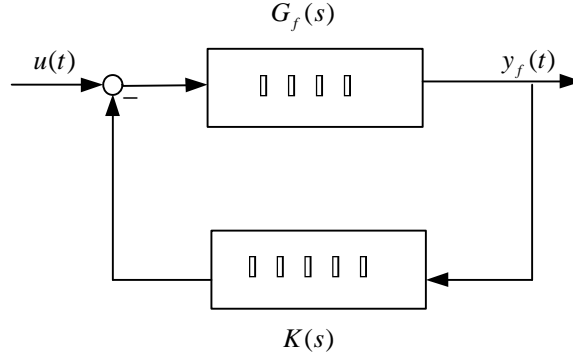


图 4-4 采用反馈补偿器的扩展系统

那么闭环传递函数矩阵为 $H(s) = C_f [sI - A_k]^{-1} B_f$ ，其中 $A_k = A_f - B_f K C_f$ 。如果存在实正定对称矩阵 P 和正实矩阵 Q ，保证下列 2 个方程式成立：

$$P A_k + A_k^T P = -Q < 0 \quad (4-55)$$

$$Q P B_f = C_f^T \quad (4-56)$$

那么闭环传递函数矩阵 $H(s) = C_f [sI - A_k]^{-1} B_f$ 为严格正实的。

值得注意的是，我们希望 $K(s)$ 取值越小越好，当 $K(s)$ 取值很小时， $H(s) \approx G_f(s)$ ，Kaufman 于 1999 年证明了采用常值输出反馈就可以使闭环系统具有严格正实性。

根据上面的讨论，以飞行系统作动器故障为例，考虑作动器 50% 以内的损伤设计了一种鲁棒性的反馈补偿器设计方案：

步骤 1：对应于 B_f ，选择 Q 矩阵，计算满足式 (4-56) 的 P 矩阵；

步骤 2：将 P 值代入式 (4-55)，求解满足不等式方程的 A_k ，进而找寻满足条件的 K ；

步骤 3：重复上面的步骤，对于所有 B_f ，求解满足正实性条件的 K 交集；

步骤 4：采用优化指标 $J = \min \|K\|$ ，选择最优反馈补偿器；

4.4 简单自适应容错飞行控制系统设计

飞机发生故障后，我们再无具体故障信息的前提下，希望尽可能的恢复被控系统的飞行品质，这要求容错控制算法具有较强的自适应能力，为此本章采用简

单自适应算法应用于容错飞行控制律设计。该设计方案的基本思想就是：选定满足性能指标的理想参考模型；通过增加合适的前馈环节或者反馈环节设计扩展矩阵，保证被控对象满足正实性条件；最后设计简单自适应增益调整方案，保证系统输出跟踪理想模型输出，从而恢复理想的输出特性，实现容错控制设计要求。容错控制系统的结构框图如下：

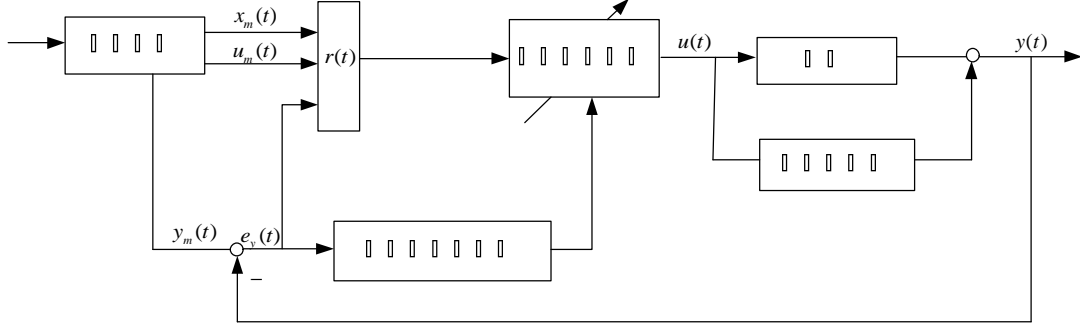


图 4-5 采用前馈补偿器的简单自适应容错控制

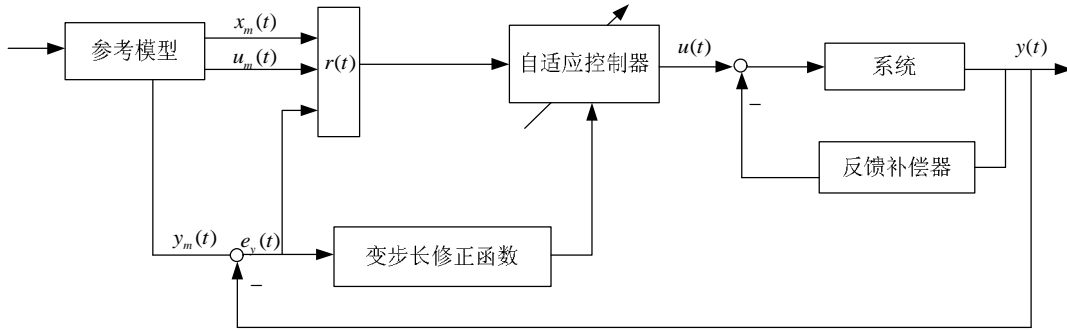


图 4-6 采用反馈补偿器的简单自适应容错控制

4.5 仿真算例

以某型飞机在 $H = 5000m, Ma = 0.5$ 的飞行状态为例验证本章提出的简单自适应容错控制方法，飞机的运动方程为：

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx \end{cases}$$

$$\text{式中, } A = \begin{bmatrix} -0.36253 & 7.237 & -0.0231 \\ -0.9975 & -0.343 & 0.0326 \\ 0.14567 & -55.252 & -2.8004 \end{bmatrix}, B = \begin{bmatrix} 0 & 0.0029 & 0 \\ -0.03703 & 0.03232 & 0.8162 \\ -0.06599 & -0.1649 & -0.1583 \end{bmatrix},$$

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}。$$

状态向量 $x = [b \quad w_x \quad w_y]^T$ ，控制向量 $u = [d_x \quad d_y \quad d_z]^T$ ， b, w_x, w_y 分别表示侧滑角、滚转速率和偏航速率改变量； d_x, d_y, d_z 分别为副翼、方向舵和尾翼的舵偏角改变量。

根据飞行品质要求选取理想的参考模型，时间常数选为 0.5，阻尼系数选择为 0.707，自然频率选为 2.828，相应的理想参考模型的状态方程描述为：

$$\begin{bmatrix} \dot{b} \\ \dot{w}_x \\ \dot{w}_y \end{bmatrix} = \begin{bmatrix} -0.1 & 0 & 1 \\ 0 & -2 & 0 \\ -8.2 & -4.2 & 0 \end{bmatrix} \begin{bmatrix} b \\ w_x \\ w_y \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ -10 & 0 & 0 \\ 0 & -10 & 0 \end{bmatrix} \begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix}$$

为了验证文中提出的控制策略的有效性，将本章的控制策略应用于跟踪滚转角速率仿真中。仿真时间为 10s，采样时间为 0.01s。3 个舵回路均为一阶惯性环节 $\frac{20}{s+20}$ ，仿真中考虑了副翼、方向舵和尾翼偏转角 $\pm 25^\circ$ 的位置饱和限制和偏转角速度 $\pm 75^\circ/s$ 的限制。

设计时考虑作动器损伤在 50% 以内。为了保证系统的严格正实性，采用反馈补偿器构成扩展系统，经过优化设计，反馈补偿增益矩阵为：

$$K = \begin{bmatrix} 15.0221 & -14.9913 & -0.3026 \\ 16.06 & -17.8133 & -0.3181 \\ 4.9044 & -9.1852 & -0.1118 \end{bmatrix}；$$

在简单自适应容错控制器部分，选取 $w_s = 3.2, w_e = 0.8$ 。正常工作状态下和故障状态下，系统采用同一个控制器结构，控制增益由自适应算法实时调整。系统在正常工作状态下的响应曲线如图 4-7 所示：

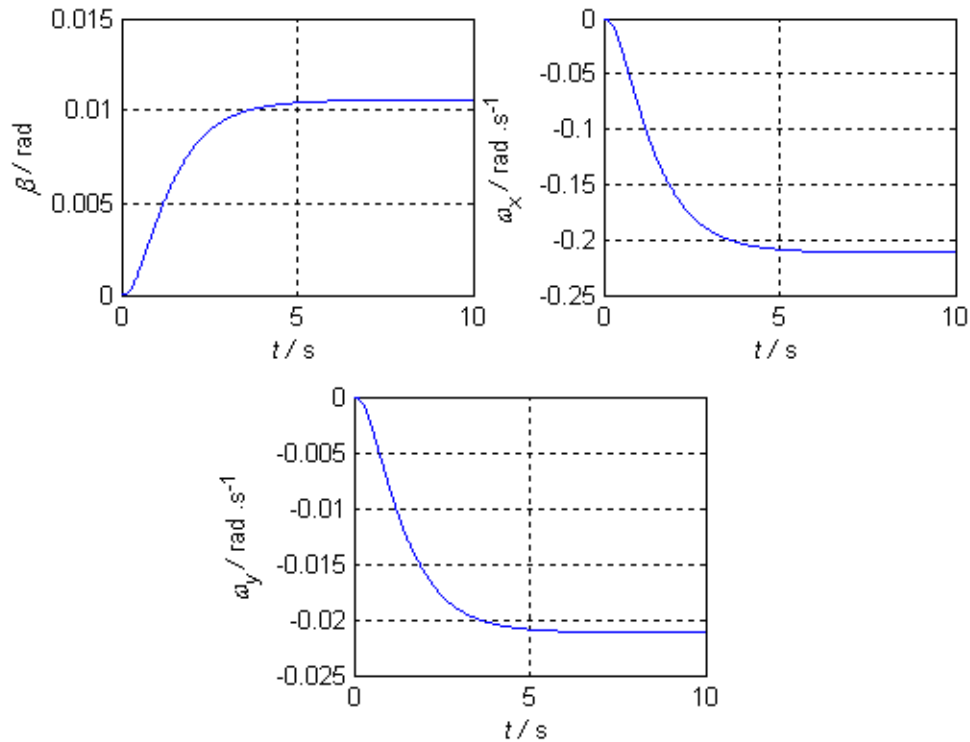


图 4-7 正常工作状态下的响应曲线

可以看出,简单自适应控制器在系统正常的工作状态可以实现对信号的良好跟踪,无超调,调节时间短。

当副翼损伤 50% 时,简单自适应容错控制器自适应调整控制增益,得到的系统响应对比曲线如下图所示:

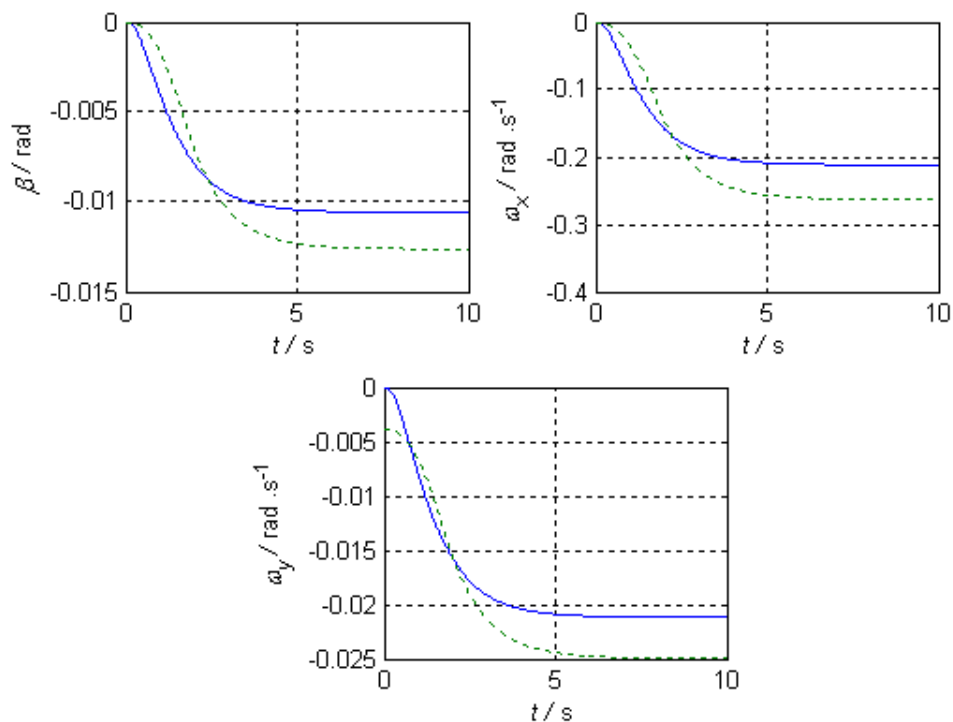


图 4-8 副翼损伤 50% 时的响应对比曲线

当方向舵损伤 50% 时系统的响应曲线为：

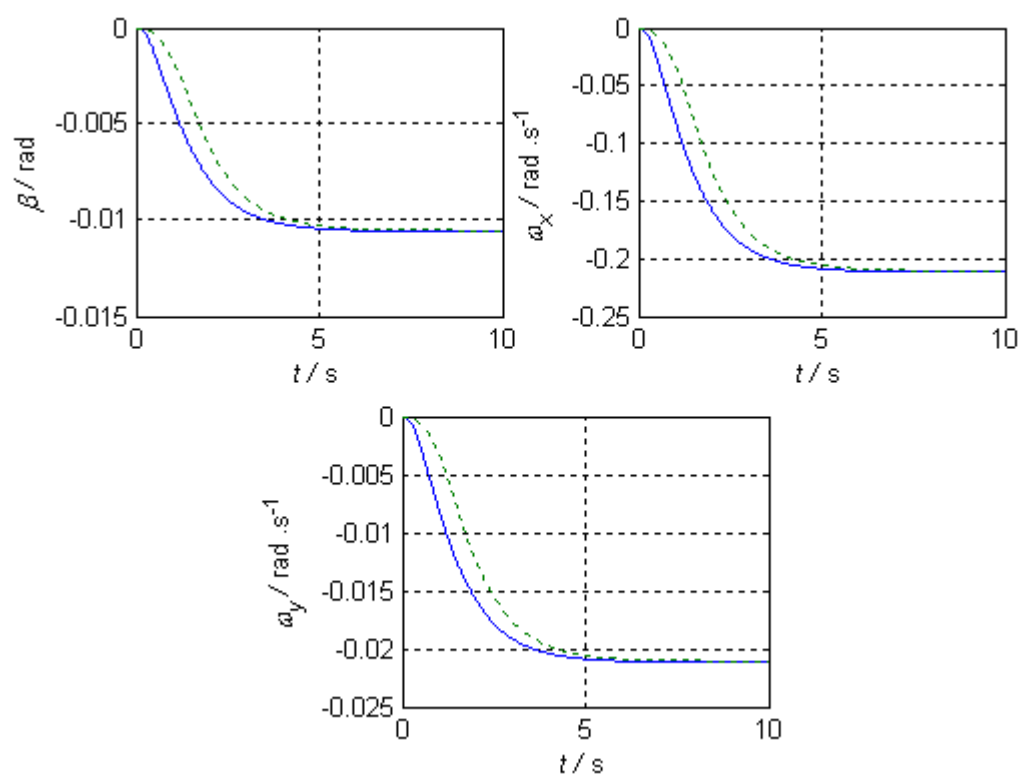


图 4-9 方向舵损伤 50%时的对比响应曲线

尾翼损伤 50%时的响应曲线如下图：

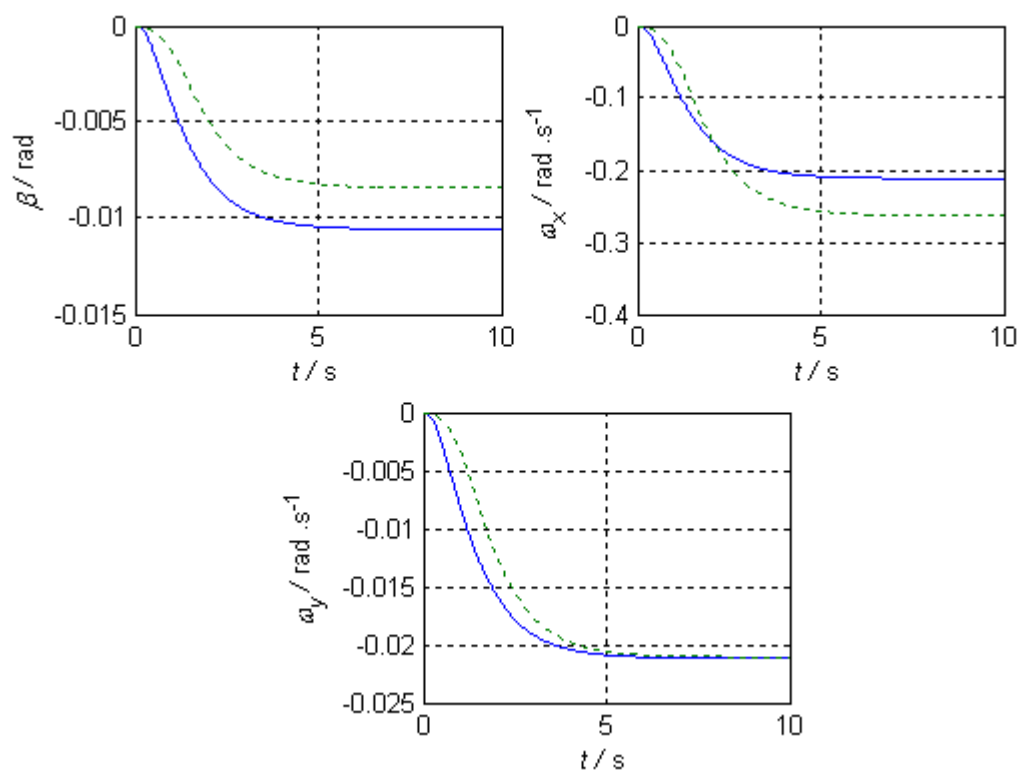


图 4-10 尾翼损伤 50%时的对比响应曲线

由对比响应曲线可以看出，副翼故障对滚转速率、偏航速率和侧滑角影响都

比较大,尾翼故障主要影响滚转速率和侧滑角,相比而言方向舵故障对系统的影响最小;不管是哪种故障状况,采用本章提出的简单自适应设计方法对故障系统进行容错控制,都可以使得故障系统具有稳定性和较好的动态性能,保证系统输出跟踪理想参考模型的输出,容错控制达到了满意的效果。

4.6 小结

本章在模型跟随自适应的基础上,讨论了简单自适应容错控制方法。该设计方案无需具体的飞行系统故障信息,设计简单,方法固定,非常适用于飞行系统的容错控制。本章分别采用前向补偿器和反馈补偿器保证了系统的严格正实性;在自适应控制器部分,引入误差积分抑制外部扰动,改进了简单自适应控制器的鲁棒性,并提出一种变步长修正自适应增益的方法设计了容错控制律,仿真结果表明在满足正实性的前提下,采用简单自适应容错控制,故障系统的输出特性可跟随理想参考模型的输出。

第五章 基于神经网络辨识器的容错飞行控制设计

由于现代飞机系统结构日趋庞大，飞行任务日益复杂，飞行环境相对恶劣，保证飞行安全和飞行品质成为目前飞行控制系统的关注重点，多种容错控制技术得到研究和发展，并在工程应用中取得了阶段性研究成果。但是目前绝大多数的容错控制方法都要求精确的故障模型和故障参数信息，这一缺点大大限制了该方法在容错控制中的应用；同时由于故障可能导致出现的系统非线性性，使得基于线性模型和控制理论的重构设计方法失效。而神经网络本质上是非线性系统，能够充分逼近任意复杂系统，应用神经网络建立系统的辨识器模型，可以在不知道具体故障信息的情况下，实现对故障环节的隔离和重构，保证飞行的安全性和可靠性。本章结合观测器控制理论，将神经网络应用于某飞机的容错控制律设计中。

5.1 引言

目前，飞机系统日趋庞大，执行飞行任务时可能面对多种复杂的不确定故障问题，这样就要求容错控制系统应具备较强的自适应性和鲁棒性，从而保证飞行系统具有适应未知故障和损伤的能力，保障飞行安全性和维持较理想的飞行品质。神经网络是对人脑或者自然神经系统若干特性的抽象和模拟，包含了大脑神经的许多特征，通过自学习，神经网络在理论上可以以任意精度逼近学习对象，具有高度的自适应性和自组织性，能够学习和适应严重不确定性系统的动态特性，并且系统信息等势分布存储在网络的各神经元及其连接权中，因此有很强的鲁棒性和容错能力；这些优点使神经网络成为容错控制领域研究的热点，引起了广大学者的关注。

神经网络的研究始于 1943 年，至今已有 60 多年的历史。心理学家 McCulloch 和数学家 Pitts 提出第一个神经模型，该模型被称为 McCulloch-Pitts (MP) 神经元模型，从此开创了神经网络理论研究的时代。1949 年，心理学家 Donald O. Hebb 提出了修正神经元连接强度的“突触修正假设”，并据此提出了神经网络的学习准则，这就是著名的 Hebb 学习准则，Hebb 的研究成果为神经网络的学习算法奠定了基础，促进了神经网络研究的早期热潮。1982 年，美国加州工学院生物物理学家 Hopfield 提出 Hopfield 网络模型，引入网络能量函数概念，并对该模型给出了稳定性判据，这标志着神经网络研究第二次高潮的到来，从此揭开了神经网络计算机研究的序幕。1986 年，E. Rumelhart 和 L. McClelland 提出了一种多层

前馈网络的反向传播学习算法^[58]，即著名的 Back-Propagation (BP) 算法，这是迄今为止影响最大的网络学习算法，该算法证明了多层神经网络具有很强的学习能力。1988 年，Cybenko 等人从理论上证明了三层神经网络具有以任意精度逼近任意连续非线性函数的能力，即神经网络的网能逼近特性^[59]。这些神经网络理论的提出为神经网络在系统控制中的研究和应用打下了坚实的基础。

对于现代容错飞行控制而言，神经网络的巨大吸引力主要在于^[60,81]：神经网络本质上是非线性系统，能够充分逼近任意复杂的非线性系统；神经网络具有高度的自适应性和自组织性，能够学习和适应严重不确定性系统的动态特性；系统信息等势分布存储在网络的各神经元及其连接权中，故有很强的鲁棒性和容错能力；信息的并行处理方式使得快速进行大量运算成为可能。

这些优点说明神经网络在解决高度非线性和严重不确定性系统的控制方面具有较大的潜力。目前，基于神经网络的容错控制已成为控制理论研究的热点之一，引起了广大学者的关注。

将神经网络应用于容错控制设计中，主要有以下两种方式：一种是利用神经网络拟合系统的正常特性，得到系统的辨识模型，构成各种控制结构。另一种方式是直接作为控制器，神经网络直接用作控制器可分为两种方式：单独作为控制器以及与其它控制律共同构成控制器。本章主要研究第一种方式在容错控制中的应用。

BP 神经网络具有结构简单、可操作性强、能模拟任意的非线性输入/输出关系等优点，是目前应用最为广泛和成功的神经网络之一。针对传统方法存在的对初始值较敏感、容易陷入局部极小点、收敛速度慢等缺点问题，本章提出了一种基于蚁群信息素思想和改进粒子群的神经网络学习算法，并结合自适应观测器理论，对故障飞行控制系统进行重构控制设计，最后将本算法应用于某型飞机的容错飞行控制律设计中进行仿真验证。

5.2 神经网络学习算法研究

5.2.1 BP 神经网络的基本结构和学习算法

1986 年，E. Rumelhart 和 L. McClelland 提出了一种多层前馈网络的误差反向传播学习算法，即著名的 Back-Propagation (BP) 算法。这是一种多层网络的逆推学习算法，其设计思想是 BP 网络的学习过程由正向传播和反向传播两部分构成。在正向传播过程中，输入样本从输入层传入，通过隐单元处理后传到输出层，每一层神经元的状态只影响下一层的神经元；如果输出层不能得到期望输出，

即期望输出与实际输出之间存在误差时，则转向反向传播过程，将误差信号沿着原来的路径返回，返回的过程中，通过逐一修改各层神经元的权值，逐次地向输入层传播进行计算，修改以后的权值再经过正向传播过程，将期望输出与实际输出比较。这两个过程不断反复运用，使得误差最小，权值不断调整的过程也就是BP神经网络的学习过程，直至达到所允许的误差范围，或者进行到设定的迭代次数为止。

BP神经网络的基本结构如图5-1所示，网络由输入层、隐含层和输出层节点组成。其中隐含层可以是一层，也可以是多层；前层节点到后层节点通过权值连接。

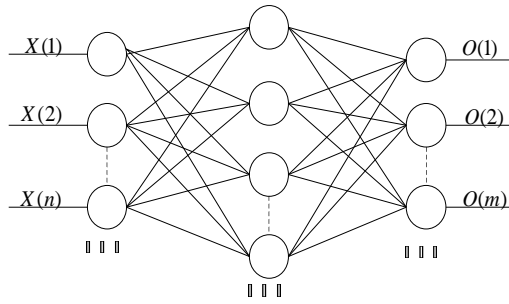


图 5-1 BP 神经网络结构图

BP神经网络的学习训练需要确定的参数包括：神经元个数、输入层到隐含层的连接权值、隐含层到输出层的连接权值以及阈值。

设有 L 层的BP网络， $(x_{1k}, x_{2k}, \dots, x_{nk})$ 表示第 k 个样本的输入， $(y_{1k}, y_{2k}, \dots, y_{mk})$ 表示第 k 个样本的输出， $(y_{1k}^d, y_{2k}^d, \dots, y_{mk}^d)$ 为相应的理想输出， $k=1 \dots N$ 表示学习样本序号， O_j^l 为第 l 层神经元 j 的输出， q_j^l 为第 l 层神经元 j 的阈值， net_{jk}^l 为第 l 层神经元 j 的输入， $f(x)$ 为非线性Sigmoid函数，则有：

$$net_{jk}^l = \sum W_{ji}^l O_{ik}^{l-1} + q_j^l \quad (5-1)$$

$$O_{jk}^l = f(net_{jk}^l) \quad (5-2)$$

对于输入层和输出层，本层输出可分别表示为：

$$O_{jk}^l = x_{jk}, j=1, 2, \dots, n \quad (5-3)$$

$$O_{jk}^l = y_{jk}, j=1, 2, \dots, m \quad (5-4)$$

以网络输出 y_{jk} 和期望输出 y_{jk}^d 之间的误差为目标函数，神经网络学习的均方

差定义为：

$$E = \frac{1}{2} \sum_j (y_{jk} - y_{jk}^d)^2 \quad (5-5)$$

采用梯度下降法调整连接权重 W_{ji}^l 和阈值 q_j^l ，令 $d_{jk}^l = -\frac{\partial E}{\partial net_{jk}^l}$ ，由式(5-1)、

(5-2)和(5-5)可得：

$$\frac{\partial E}{\partial W_{jk}^l} = \frac{\partial E}{\partial net_{jk}^l} \cdot \frac{\partial net_{jk}^l}{\partial W_{jk}^l} = -d_{jk}^l O_{jk}^{l-1} \quad (5-6)$$

输出层的训练误差为：

$$d_{jk}^l = y_{jk} (y_{jk}^d - y_{jk}) (1 - y_{jk}) \quad (5-7)$$

输出层依次向隐含层反向分配的误差为：

$$d_{jk}^l = O_{jk}^l (1 - O_{jk}^l) \sum_m d_{km}^{l+1} W_{jm}^{l+1} \quad (5-8)$$

得到权值与阈值的修正式为：

$$W_{ji}^l(t+1) = W_{ji}^l(t) + a d_{jk}^l O_{ji}^{l-1} \quad (5-9)$$

$$q_j^l(t+1) = q_j^l(t) + a d_{jk}^l \quad (5-10)$$

上式中， a 为学习步长， $0 < a < 1$ ， l 为迭代次数。

采用式(5-9)与(5-10)反复迭代，直到对所有的学习样本网络输出的均方误差达到要求为止，即

$$E = \frac{1}{2N} \sum_{k=1}^N E_k \leq e \quad (5-11)$$

从上面的推导中可以看到，在计算多层前馈网络的训练误差时，是从输出层开始，逐层向后进行的，即误差反向传播。在整个训练过程中，输入样本周期性进入网络，直至网络收敛，输出误差满足训练终止精度。

5.2.2 基于蚁群和改进粒子群的神经网络学习算法

标准的 BP 神经网络采用梯度下降法对神经网络的参数进行训练，这虽然是误差函数下降最快的方向，但不一定是算法收敛的最优方向。这是由于神经网络的误差函数曲面是多维空间中一个形状极为复杂、不规则的曲面，存在多个局部极小点，其误差梯度都为 0，这使得 BP 神经网络在训练过程中，很容易陷入局

部极小点而不能自拔；并且在误差曲面上有一些区域平坦，此时误差对于权值的变化不敏感，误差梯度变化小，调整时间长，收敛速度慢；同时采用梯度下降法训练神经网络时，网络的参数初始值是在一定范围内随机产生的较小值，在随机产生的初始值下，网络训练从误差曲面的某个初始点开始，沿曲面下降的方向滑向曲面的某个“谷底”，这个“谷底”可能是全局最小点，也可能是局部极小点，因此，BP 算法对网络的初始参数异常敏感，不同的初始权值和阈值可能导致完全不同的结果。

针对传统方法存在的对初始值较敏感、容易陷入局部极小点、收敛速度慢等缺点问题，国内外不少学者提出了许多改进算法。Johnsone E.M^[105]提出了一种共轭梯度算法，采用共轭梯度的方向进行直线搜索，实现每步迭代在搜索方向上获得最优步长，保证目标函数递减，算法的收敛速度加快了许多，一定程度上解决了网络学习速度慢的问题。Montana D J^[106]采用遗传算法对神经网络的连接权值进行优化，这是一种模拟生物进化过程的随机优化算法，具有良好的全局搜索能力，无需计算误差函数的梯度信息就可以进化学习到问题的接近最优解，利用遗传算法的全局寻优能力来获取最佳网络权值，提高了神经网络的收敛速度和学习能力。Jarmo Ilonen^[107]等采用差异演化算法训练前向神经网络，运用差异演化算法的并行寻优优势，克服了梯度下降法易陷入局部极小值的问题，提高了网络的训练速度。

这些改进大多集中在权值训练上，对神经网络的结构并没有进行过多的研究，而在神经网络的设计中，隐含层神经元节点数也关系到网络设计的好坏。为此本章引入爬山思想改进了粒子群算法，提高了粒子群的效率和解的质量，以此训练神经网络的权值；在此基础上结合蚁群算法，提出一种并行学习方法对神经网络的结构和权值同时进行训练。该算法的基本思想是：对隐层节点设置选择因子，借鉴蚁群算法中蚁群释放信息素的思想，在隐层节点的选择因子中释放信息素，通过更新信息素达到调整神经网络隐层结构的目的；同时在当前结构下采用改进粒子群学习网络权值。这种策略充分利用了群体的搜索经验，实现了优化过程中不同结构间信息的共享，从而完成神经网络结构和权值的并行优化。

5.2.2.1 基于爬山策略改进的粒子群算法

粒子群优化算法源于鸟群捕食行为的启发，是一种基于迭代的群智能优化工具，具有流程简单、算法简洁、全局优化等特点。

应用粒子群算法在训练神经网络参数问题时，问题的解对应于空间 D 中的一个粒子，每个粒子都有自己的位置和速度，在算法实现过程中以粒子对解空间中最优粒子的追随进行解空间的搜索，粒子通过跟踪两个极值来更新自己，一个是

粒子本身找到的最优解- $pbest$ ，即个体极值；另一个是整个种群找到的最优解- $gbest$ ，即全局极值。基于爬山策略改进的粒子群算法，其基本思想是对于每代的全局极值，通过领域搜索进行爬山操作，从而提高粒子群搜索的效率和解的质量。其算法实现如下所示：

表 5-1 基于爬山策略改进的粒子群算法

步骤 1，确定神经网络的输入节点数 m 、输出节点数 n 和隐层节点数 h ，以及网络训练的目标函数；

步骤 2，初始化：确定粒子群中粒子的维数 $D = m \times h + h + h \times n$ 以及群体规模 N ，设定粒子群优化算法的参数，包括惯性权重的初始值 w_s 和终值 w_e 、加速系数 c_1 和 c_2 、速度的限定值 V_{\max} ，粒子群算法的最大迭代次数 T 或迭代终止精度 e ，在 $[0, 1]$ 内随机初始化粒子的位置和速度，并且设定爬山算法的更新次数 l ；

步骤 3，将粒子的位置向量解码成对应的网络参数，计算神经网络的均方误差，得到粒子群的适应度函数。根据式（3-56）、（3-57）计算粒子的个体极值 $pbest$ 和全局极值 $gbest$ ；

步骤 4，对全局极值 $gbest$ 通过领域搜索实施爬山操作，更新全局极值；

步骤 5，根据式（3-54）对粒子进行速度和位置的更新，若粒子的速度超出限定值 V_{\max} ，则速度取边界值；

步骤 6，检验粒子群算法是否满足终止条件，若满足，则停止迭代，输出最优解；否则转到步骤 3，进行下一个迭代周期的训练。算法的终止条件为粒子群算法达到最大迭代次数 T 或者粒子目标函数达到了给定的迭代终止精度 e ，记录此时得到的最优解。

5.2.2.2 蚁群算法

在神经网络的设计中，隐含层神经元节点数关系到网络设计的好坏，隐含层神经元数目过少会导致网络的容错性和自适应性降低，特别是处理较复杂的问题时容易导致神经网络难以训练出理想的结果；而隐含层神经元过多则会增加网络的学习时间。传统的训练算法一般根据经验公式或者实验的方法来确定隐含层节点的个数，带有很大的盲目性；针对神经网络隐含层节点数确定盲目、缺乏系统的理论指导这一问题，本章采用蚁群算法中的信息素思想和爬山策略改进的粒子群算法相结合，提出一种网络结构和权值并行调整的神经网络学习算法。

蚁群算法是意大利学者 Marco Dorigo 受到现实世界中蚂蚁觅食行为的启发

而提出的^[164]，在蚂蚁进行觅食路径选择时，蚂蚁个体间通过“信息素”进行信息传递，蚂蚁都倾向于朝着信息素强度高的方向移动。每一只蚂蚁完成一个循环后，对各条路径上信息素量进行加强或挥发调整。经过一定的循环次数，经过蚂蚁越多的路径积累的信息素就越多，后面的蚂蚁选中该路径的概率也就越大；而某些路径上通过的蚂蚁较少，其路径上的信息素就会随时间的推移而挥发掉，这种挥发机制可以使蚂蚁逐渐忘记过去、不受以往经验的过分约束。在蚁群算法中，每只蚂蚁根据路径上的信息素强度来选择状态转移方向，信息素量和状态转移规则对算法性能具有重要意义。

蚁群算法中，初始时刻所有路径上的信息素量相等，蚂蚁完成一次循环以后，路径 (i, j) 上信息素的更新规则为：

$$t_{ij}(t+1) = (1-r)t_{ij}(t) + \Delta t_{ij}(t, t+1) \quad (5-12)$$

$$\Delta t_{ij}(t, t+1) = \begin{cases} \frac{Q}{L_g} & \text{若}(i, j) \in \text{本次循环全局最优路径} \\ 0 & \text{否则} \end{cases} \quad (5-13)$$

(5-12)、(5-13)式中， $t_{ij}(t+1)$ 为在 $t+1$ 时刻路径 (i, j) 上的信息素量， $\Delta t_{ij}(t, t+1)$ 表示蚁群在 $t+1$ 时刻留在路径 (i, j) 上的信息素增量， $0 < r < 1$ 称为信息素挥发因子，控制路径上信息素的挥发速度，避免路径上的信息素量的无限累加， $1-r$ 称为信息素残留因子； Q 称为信息素强度，控制每次释放信息素的数量； L_g 为到目前为止找出的全局最优路径的长度，即目标函数。每次循环以后，只有本次循环中全局最优路径的蚂蚁才会释放信息素，而非全局最优路径上的信息素则逐渐挥发。

蚁群算法中，各条路径上的信息素量决定了蚂蚁的转移方向。 t 时刻，蚂蚁由当前路径点 i 向下一个路径点 j 移动时，对各可行路径的选择概率即蚂蚁的状态转移规则为：

$$p_{ij}(t) = \begin{cases} \frac{[t_{ij}(t)]^a [h_{ij}(t)]^b}{\sum_{u \in J(i)} [t_{iu}(t)]^a [h_{iu}(t)]^b} & \text{若 } j \in J(i) \\ 0 & \text{否则} \end{cases} \quad (5-14)$$

其中， t_{ij} 为路径 (i, j) 上的信息素强度， h_{ij} 为路径 (i, j) 的能见度因数，反映了路

径 (i, j) 的上启发信息, 与蚁群处理的具体问题有关, 不随着蚁群系统的运行而改变, a 和 b 为调节参数, 反映了蚂蚁在运动过程中所积累的信息素和启发信息, $J(i)$ 为路径点 i 的所有可行路径集合。蚁群算法的状态转移规则称为随机比例规则。

借鉴蚁群算法的信息素思想, 本文采用蚁群算法中基于群智能的路径选择机制应用于神经网络的结构调整, 并结合改进的粒子群算法, 提出一种神经网络学习算法。

5.2.2.3 基于蚁群和改进粒子群的神经网络学习算法

针对传统学习方法中神经网络结构的调整有很大盲目性, 本文在神经网络的学习过程中对隐层结构中的每个节点引入选择因子, 因子选“0”, 相当于该节点不存在; 因子选“1”, 相当于该节点存在, 选择因子对应于蚁群算法中的路径, 每个选择因子对应于二维的信息素。在当前结构下, 通过爬山策略改进的粒子群优化算法找到神经网络的最优权值, 同时完成对每个节点对应的蚁群路径信息素的更新。值得注意的是, 由于目前对神经网络的信息处理机制的了解比较肤浅, 神经网络隐层中各神经元的具体物理意义还不是很清楚, 因此在本算法中, 蚁群的选择概率公式中并不存在启发性信息。

采用蚁群策略和改进粒子群并行学习神经网络的结构和权值, 可以实现优化过程中不同网络结构间信息的共享, 从而改善神经网络结构调整的盲目性, 引入信息素的思想对下步网络结构的调整具有一定的指导作用。

根据问题的不同, 神经网络的训练方式可分为离线处理和在线学习, 本章采用离线处理的方式训练神经网络。离线方式是指给定数据样本集, 在每个迭代周期, 根据样本数据和网络输出计算误差值, 对网络参数进行更新, 直到满足迭代终止精度。其训练过程可分为初始化、迭代计算和返回结果这几个环节, 基于蚁群和改进粒子群的神经网络学习算法如下:

步骤 1: 确定神经网络训练的目标函数, 网络的输入节点数 m 、输出节点数 n 和隐层选择因子 $N = (n_1, n_2 \dots n_N)$, $n_i = 0$ 或者 1, n_i 对应于一个二维的信息素 (t_i^0, t_i^1) , 以及蚁群算法的循环次数 k ;

步骤 2: 根据蚁群的随机转移概率 (5-14), 选择一组选择因子, 即得到对应的神经网络隐含层结构;

步骤 3: 确定粒子群中粒子的维数 $D = m \times h + h + h \times n$, $h = \sum_N n_i$ 和群体规模 N , 设定粒子群优化算法的参数, 包括惯性权重的初始值 w_s 和终值 w_e 、加速系

数 c_1 和 c_2 、速度的限定值 V_{\max} ，粒子群算法的最大迭代次数 T 或迭代终止精度 e ，在 $[0, 1]$ 内随机初始化粒子的位置和速度，并且设定爬山算法的更新次数 l ；

步骤 4：将粒子的位置向量解码成对应的网络参数，计算神经网络的均方误差，得到粒子群的适应度函数。根据式（3-56）、（3-57）计算粒子的个体极值 $pbest$ 和全局极值 $gbest$ ；

步骤 5：对全局极值 $gbest$ 通过领域搜索实施爬山操作，更新全局极值；

步骤 6：根据式（3-54）对粒子进行速度和位置的更新，若粒子的速度超出限定值 V_{\max} ，则速度取边界值；

步骤 7：检验粒子群算法是否满足终止条件，若满足，则停止迭代，输出最优解；否则转到步骤 4，进行粒子群下一个迭代周期的训练。算法的终止条件为粒子群算法达到最大迭代次数 T 或者粒子目标函数达到了给定的迭代终止精度 e ，记录此时得到的最优解即当前结构下神经网络网络的最优参数值；

步骤 8：记录当前的 $N = (n_1, n_2 \dots n_N)$ 、神经网络权值矩阵 w_i 和相应的目标函数 J_i ，并根据公式（5-12）、（5-13）由当前目标函数 J_i 对隐层选择因子 $N = (n_1, n_2 \dots n_N)$ 中的每一个 n_i 更新其对应的信息素 (t_i^0, t_i^1) ；

步骤 9：判断是否达到蚁群循环次数，若满足，则停止更新，转入步骤 10；若不满足，则转到步骤 2，继续进行蚁群搜索；

步骤 10：令 $J = \text{mix}(J_1, J_2 \dots J_k)$ ， J 对应的神经网络权值矩阵即最终得到的最优权值，相应的 $N = (n_1, n_2 \dots n_N)$ 即最终得到的最优结构。

5.3 基于神经网络辨识器的容错控制设计

5.3.1 神经网络辨识原理

系统辨识实质为优化问题，方法大体可分为两大类，一种是基于算法的，如最小二乘法、梯度校正法、极大似然法；一种是基于神经网络的。基于算法的辨识方式通过建立模型，将辨识问题转化为模型参数估计问题，适用于线性系统。飞机出现故障后，系统的参数特性发生改变，系统可能出现非线性性，在这种情

况下可能导致基于算法的线性辨识方法失效^[66]。

而神经网络具有辨识非线性系统的能力，基于神经网络的辨识方法就是将系统辨识问题转化为对某个非线性函数的逼近问题。其基本思想是采用神经网络逼近被测系统，利用神经网络具有以任意精度逼近任意非线性函数的能力，通过学习被测系统的输入/输出，使得系统输出和网络输出的误差达到最小，从而实现系统的辨识。神经网络训练的过程即对系统的辨识过程，训练好的神经网络在外部特性上与被测系统等价。神经网络辨识结构如图 5-2 所示，其中神经网络与被辨识系统具有相同的输入，二者输出误差作为网络的训练信号。学习结束后，神经网络模型与实际系统具有相同的输入输出映射特性，因此可以作为系统的一个物理冗余通道。

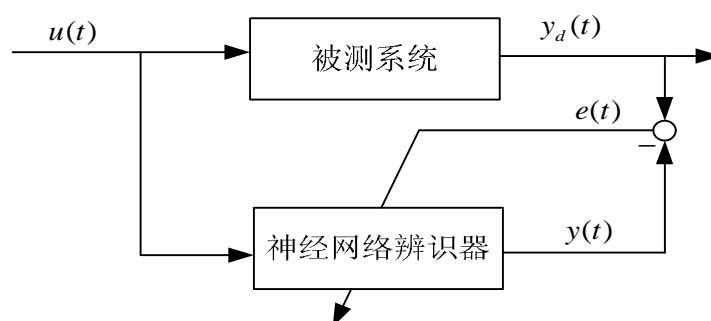


图 5-2 神经网络辨识结构图

使用神经网络进行系统辨识的理论基础是函数逼近理论，【定理 5.1】说明只含一层隐含层的前向网络具有逼近任意非线性函数的能力，是一个通用的逼近器。多层前向网络具有极强的逼近能力，如果仅含一个隐含层的前向神经网络可以逼近某个函数，那么三层以上的网络也具备同样的能力，其最佳逼近能力使神经网络成为非线性系统辨识的有力工具。

基于神经网络辨识主要有以下四大特点：

（1）辨识器由神经网络构成，其本质上是非线性系统，具有逼近任意非线性函数的能力，因此被测系统允许是非线性系统；

（2）辨识过程是非算法式的，通过神经网络的自学习，网络拟合被测系统的输入输出特性，训练好的神经网络与系统在外特性上等价，因此可以认为神经网络辨识器是被测系统的一个物理实现；

（3）辨识之前无需对被测系统建立模型，神经网络的权值对应于模型参数，反映了输入输出关系，通过调节权值就可以使得网络输出逼近系统输出；

（4）辨识算法的收敛性和收敛速度完全取决于神经网络，因此通过优化神

神经网络的结构和学习算法可以实现对辨识过程的优化。

应用神经网络进行系统辨识最早是由 Narendra 和 Patrhasarathy^[39]于 1990 年提出,并在随后的十几年里得到深入发展。1996 年 Narendra^[40]总结了 1990 年至 1995 年之间神经网络方法在系统辨识和控制领域中的研究进展和应用,指出神经网络在这些领域具有广阔的应用前景。

5.3.2 基于神经网络辨识器的自适应容错控制

由于神经网络极强的非线性拟合能力,使其在非线性系统辨识中得到了充分应用。对于被测系统,通过对神经网络进行训练可以有效的对系统的时变特性进行跟踪。基于输入/输出数据的辨识方法,在不需要了解系统的具体信息的情况下,就可以直接利用神经网络的学习功能建立精确的神经网络辨识器模型。构造出系统的辨识器之后,就可以将其用于系统的故障隔离与重构控制。

值得注意的是,建立系统的神经网络辨识器模型有两种方式,一种是离线学习,这种方式采用特定输入信号下的数据结构组成输入/输出序列样本训练神经网络,直至满足精度要求,完成对系统的辨识,对训练时间没有严格的要求;另外一种在线辨识方法,多采用串-并联结构,针对不同输入信号下的输入/输出数据,拟合系统的输出,适用性更为广泛,相应的对实时性要求较高,本章对于神经网络,采用离线训练、在线控制的方式应用于飞行容错控制系统设计之中。

将训练好的神经网络作为正常模型,网络输出与当前系统输出进行比较,得到输出残差,进一步判决被测系统是否发生故障。

5.3.2.1 残差决策原理

残差是被测系统状态的函数,包含了可能发生的故障信息,在理想情况下,残差值为零反映被测系统与神经网络描述相同,即系统实际输出与该系统正常状态下的输出相同,表示系统无故障发生,正常工作;残差值偏离零对应于系统发生了故障。而工程实际中由于外部扰动和噪声的影响,导致输出残差可能偏离故障模型偏差,即无故障条件下的残差可能不为零,故障情况下的残差可能为零,因而需要进行故障检测的残差决策,从受干扰的残差信号中正确判断故障。

残差决策有三种基本方法,一种门限判决法,当残差超过门限值则判决为故障;一种是移动窗口平均法,残差信号经过移动窗口平均法滤波后进行门限判决;还有一种就是序贯概率比假设检验(Sequential Probability Ratio test, SPRT),其原理为通过二元假设检验的概率比计算以获得最大的信噪比,然后与不同最佳判决准则的门限值对比。相对于前两种方法,概率比假设检验是最佳意义上的判决,而且这种方法并不预先规定观测样本的数目和观测时间,而是在检验过程中增加

观测数据，一直到满足判决要求，非常适合于实时监测，因此在飞行控制系统的故障判决中主要采用序贯概率比假设检验法。下面简单介绍一下序贯概率比假设检验法的基本算法。

在故障判决中， H_0 表示无故障， H_1 表示有故障，则有四种可能性：

- (1) H_0 为真，判决 H_0 成立，即系统发生故障；
- (2) H_0 为真，判决 H_1 成立，这种情况称为故障漏报，其概率为 P_M ；
- (3) H_1 为真，判决 H_1 成立，即系统无故障正常工作；
- (4) H_1 为真，判决 H_0 成立，这种情况称为故障虚警，其概率为 P_F 。

对于 N 次独立观测序列样本 $R(N) = \{r(1) \dots r(N)\}$ 建立似然比：

$$L(R) = \frac{p(R(N) | H_1)}{p(R(N) | H_0)} = \frac{p(r(1) \dots r(N) | H_1)}{p(r(1) \dots r(N) | H_0)} = \sum_N L(r(i)) \quad (5-17)$$

其中：

$$L(r(i)) = \frac{p(r(i) | H_1)}{p(r(i) | H_0)} \quad (5-18)$$

通常情况下的残差遵从正态分布，设均值为 m ，方差为 s^2 ，正常情况下残差均值记做 m_0 ，故障情况下残差均值记做 m_1 ，则式(5-18)可写为：

$$L(r(i)) = \frac{p(r(i) | H_1)}{p(r(i) | H_0)} = \ln \frac{e^{-(r(i)-m_1)^2/2s^2}}{e^{-(r(i)-m_0)^2/2s^2}} \quad (5-19)$$

定义瓦尔德判决门限 $T(H_1)$ 和 $T(H_0)$ ：

$$T(H_1) = \ln \frac{1-P_M}{P_F} \quad (5-20)$$

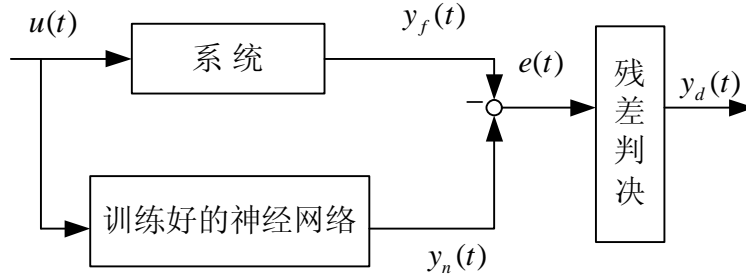
$$T(H_0) = \ln \frac{P_M}{1-P_F} \quad (5-21)$$

则 SPRT 判决规则为：

$$\begin{cases} L_N(R) \geq T(H_1): \text{判决} H_1 \text{为真;} \\ L_N(R) \leq T(H_0): \text{判决} H_0 \text{为真;} \\ T(H_0) < L_N(R) < T(H_1): \text{继续采样检验} \end{cases} \quad (5-22)$$

按照这个判决规则可以实时监测系统，将训练好的神经网络作为正常模型，

网络输出与当前系统输出进行比较，得到输出残差，进一步判决被测系统是否发生故障，若系统发生故障则采用神经网络代替被测系统，从而完成容错控制设计。



5.3.2.2 基于神经网络辨识器的容错控制策略

本章基于神经网络辨识器，结合残差序贯概率比假设检验，设计了容错飞行控制系统，具体步骤如下：

步骤 1：根据系统在某信号下的输入/输出样本集合，采用基于蚁群和改进粒子群的神经网络学习算法离线训练神经网络，学习被测系统，训练好的神经网络具有和系统相同的外部特性，可以作为系统的一个物理冗余通道；

步骤 2：通过训练好的神经网络在线判决故障，即系统的输入信号作为网络输入，比较网络输出和当前的系统实际输出，得到残差信息，判决被测系统是否发生故障；

步骤 3：若判断系统正常工作，则无需进行重构控制；若判断系统发生故障，则使用神经网络输出代替系统输出，应用于控制回路，保证正常工作状态，从而完成系统的容错控制设计。

5.4 仿真算例

为了验证控制策略的有效性，下面对某飞机的传感器故障进行仿真验证。

飞机的基本参数如下：

$$[cbar, b, s, lx, ly, lz, Jxy, Jxz, Jyz, m]$$

$$= [8.32, 59.74, 510.95, 24675560, 44876980, 67383260, 0, 1315126, 0, 288772]$$

$$[CD0, CDa, CDq, CDde, CDih] = [0.0164, 0.2, 0, 0, 0]$$

$$[CL0, CLa, CLq, CLde, CLih] = [0.21, 4.4, 6.6, 0.32, 0.7]$$

$$[Cm0, Cma, Cm q, Cmde, Cmih] = [0, -1, -20.5, -1.3, -2.7]$$

$$[CY0, CYb, CYp, CYr, CYda, CYdr] = [0, -0.9, 0, 0, -0, 0.12]$$

$$[Cl0, Clb, Clp, Clr, Clda, Cldr] = [0, -0.16, -0.34, 0.13, -0.013, 0.008]$$

$$[Cn0, Cnb, Cnp, Cnr, Cnda, Cndr] = [0, 0.16, -0.026, -0.28, -0.0018, -0.1]$$

在自动驾驶仪状态下，初始状态为：

$$[V, a, b, w_x, w_y, w_z, y, J, f, x_e, y_e, H] \\ = [150.148, 0.0464959, 0, 0, 0, 0, 0, 0.0464959, 0, 0, 0, 61]$$

选择三层神经网络离线学习正常系统，网络训练步数 $T = 1500$ ，网络的初始权值和阈值随机初始化在 $[-1, 1]$ 之间；粒子群算法的参数设置为：群体规模 $N = 25$ ，加速因子 $c_1 = c_2 = 2$ ，惯性权重按曲线调整，初始和终止惯性权值为 $w_s = 0.9, w_e = 0.2$ ；爬山算法的迭代次数 $l = 20$ ；隐层节点的信息素释放因子 $Q = 0.001$ ，信息素挥发因子 $r = 0.9$ ，最大初始隐层节点数 $N = 30$ ，训练好的神经网络输出可以作为系统的一个物理冗余通道。

以俯仰角速率传感器发生卡死、恒偏差故障为例，采用本章提出的控制策略对系统进行容错控制，仿真时间为 30s，采样时间为 0.05s，设置虚警率 $a = 0.05$ ，漏报率 $b = 0.05$ ，序贯概率比最大校验宽度为 10。考虑仿真过程中的高斯白噪声。各故障状态下的仿真检测结果如下。

设俯仰角速率传感器在第 200 个采样点发生卡死故障，卡死位置为 0.02rad/s，故障校验时间为 207 个采样点，神经网络辨识器和传感器残差输出如下：

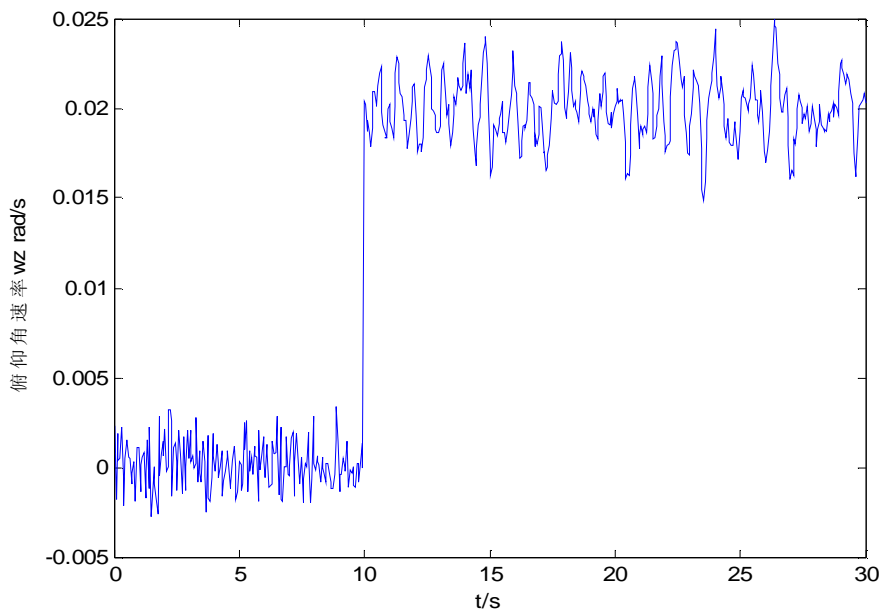


图 5-4 卡死故障时的残差曲线

可以看出,神经网络辨识器很好地拟合了正常工作状态下的传感器输出,无故障发生时,神经网络辨识器和传感器的输出残差较小;当故障出现时,输出残差产生了很大的跳动。可见神经网络辨识器对故障非常敏感,适于实时的故障诊断和定位。

故障发生后,采用神经网络输出信号代替故障传感器信号反馈给闭环系统,对系统进行重构控制,重构响应如图 5-5 所示,可以看出,神经网络辨识器重构准确及时,故障发生的瞬间,输出发生了变化,但是很快就恢复到正常状态下的输出,重构达到了满意的效果。

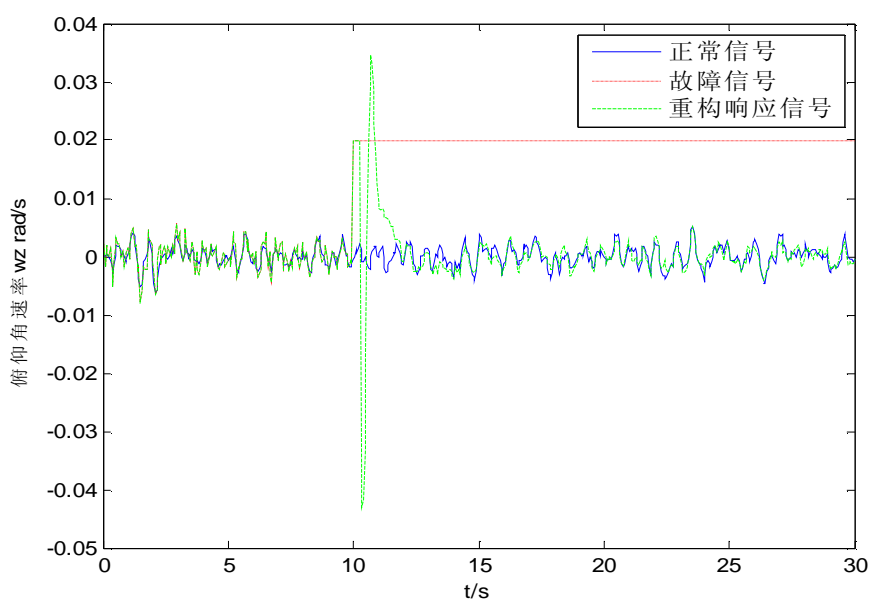


图 5-5-1 卡死故障后的俯仰角速率重构响应曲线

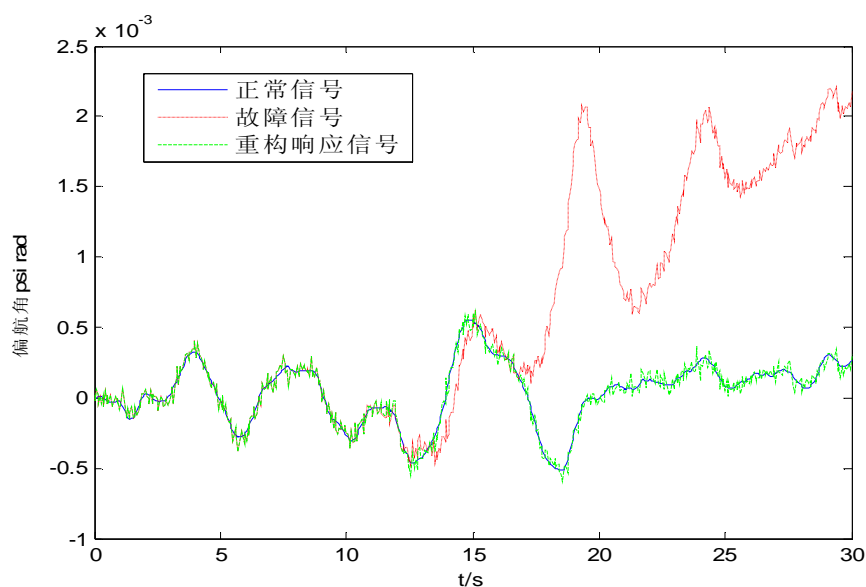


图 5-5-2 卡死故障后的偏航角重构响应曲线

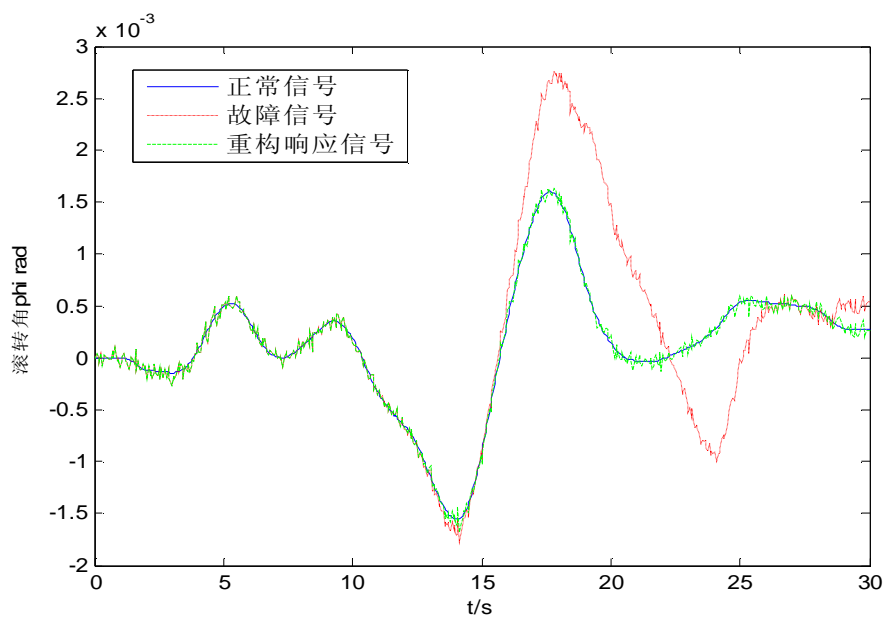


图 5-5-3 卡死故障后的滚转角重构响应曲线

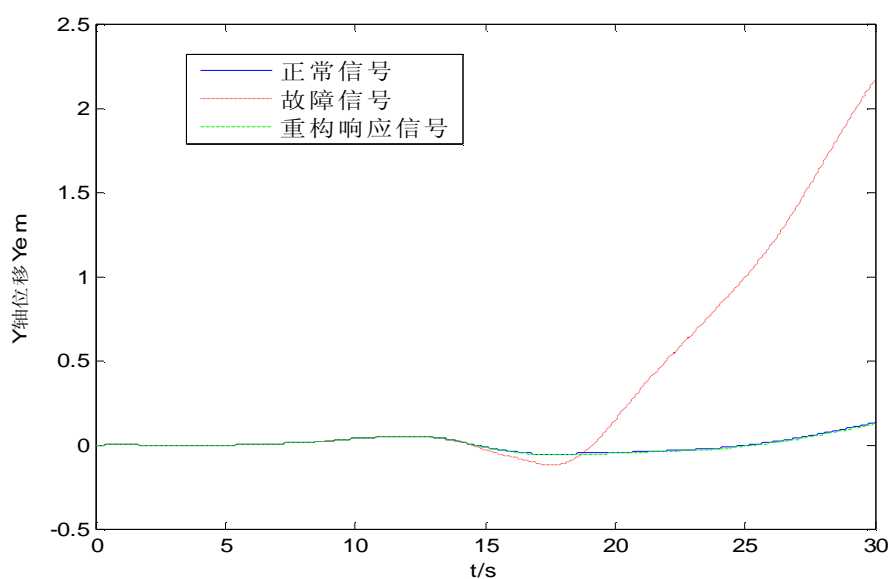


图 5-5-4 卡死故障后的 Y 轴位移重构响应曲线

下面以传感器的恒偏差故障为例，仿真验证本章的容错控制策略。

考虑俯仰角速率传感器在第 200 个采样点发生偏差 0.01rad/s 故障，故障校验时间为 212 个采样点，神经网络辨识器和传感器残差输出如下：

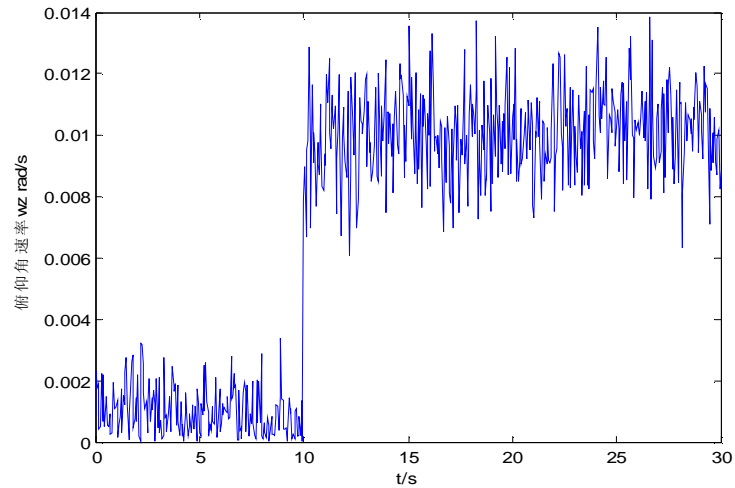


图 5-6 恒偏差故障时的残差曲线

采用神经网络辨识器对系统进行重构控制，响应如下：

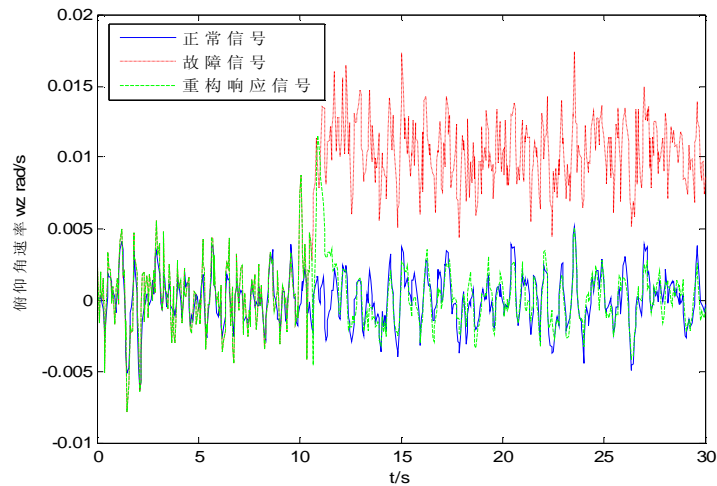


图 5-7-1 恒偏差故障后的俯仰角速率重构响应曲线

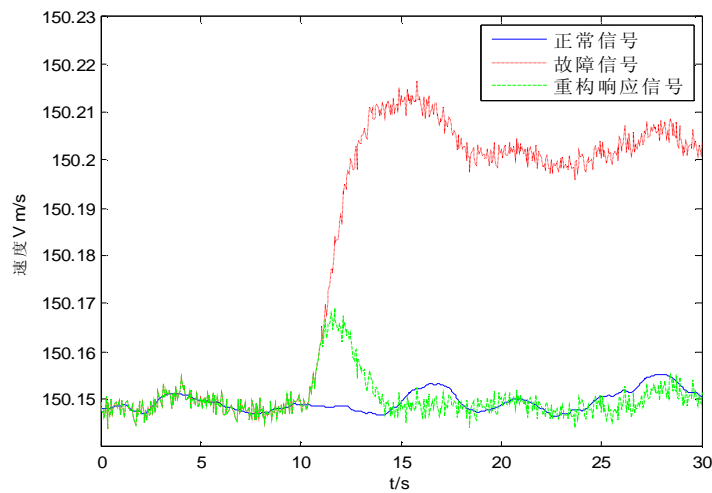


图 5-7-2 恒偏差故障后的速度重构响应曲线

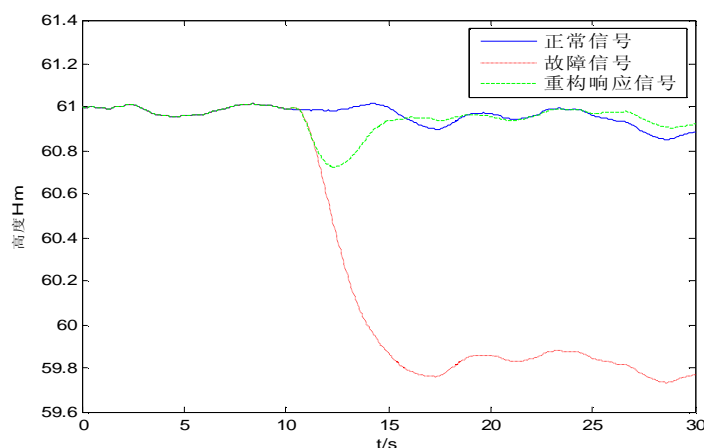


图 5-7-3 恒偏差故障后的高度重构响应曲线

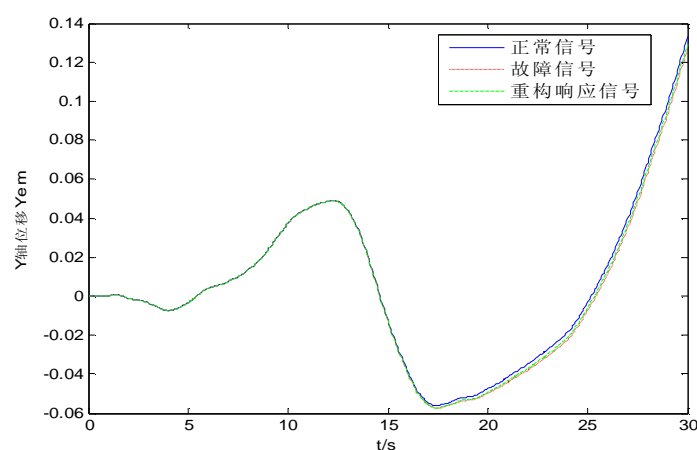


图 5-7-4 恒偏差故障后的 Y 轴位移重构响应曲线

基于神经网络辨识器的容错控制对传感器故障敏感，重构准确及时。采用这种策略对多通道的传感器分别建立对应的神经网络辨识器，可以迅速、准确地进行故障诊断和重构控制。

5.5 小结

本章主要研究了神经网络辨识器在飞行控制系统故障诊断与自适应重构中的应用，提出一种对神经网络的结构和权值并行训练的学习策略，该策略采用改进了粒子群算法学习网络权值；在此基础上结合信息素思想调整神经网络隐层结构，实现了优化过程中不同结构间信息的共享，从而完成神经网络结构和权值的并行优化。本章采用这种策略离线建立了神经网络辨识器模型，然后根据飞控系统实时、闭环的特点，结合残差序贯概率比假设检验，采用神经网络辨识器对飞控系统进行故障隔离与自适应重构设计，并对设计的算法进行仿真验证。

第六章 基于神经网络控制器的容错飞行控制设计

在上一章我们提到，由于现代飞机结构的日益庞大、飞行任务的日趋复杂、飞行环境的恶劣，保证飞机安全性和飞行品质的容错控制被提上研究日程，成为飞行控制领域的关注热点。由于故障可能引起的系统非线性性和不确定性，导致依赖精确数学模型和故障信息重构控制策略失效；而神经网络以其高度的自适应性和自组织性、能够学习和适应严重不确定性系统的特点，在解决高度非线性和严重不确定性系统的控制方面具有较大的潜力，成为容错控制的有力工具。采用神经网络方法设计容错飞行控制系统，主要是以两种方式：一种是采用神经网络作为辨识器，拟合系统的正常特性，得到系统的辨识模型，从而构成各种控制结构；另外一种是以神经网络直接或者和其他控制方法结合作为自适应容错控制器，本章着重讨论后者的应用，对于非线性不确定的飞行控制系统故障，提出一种自适应鲁棒容错飞行控制设计方法。

6.1 引言

1943 年，心理学家 McCulloch W.S 和青年数学家 Pitts W.A. 合作，提出第一个神经模型（MP），从而开创了对神经网络的研究。在神经网络方法进入控制领域之前，传统的控制律设计方案是建立数学模型，采用数学表达式描述系统，然后进行相应的数学求解。这种方法与数学模型直接相关，模型不精确和参数扰动都会影响控制效果^[66]。

神经网络控制不采用传统的设计思路，无需被控对象的数学模型，具有良好的自适应和自学习能力，神经网络的训练过程也就是控制器的调整、完善过程。目前，对神经网络的研究还处于探索阶段，神经网络控制器的设计主要分为两种方式：一种是采用神经网络单独构成控制器；一种是与其他控制方法结合构成控制器，比如对于线性部分特性已知而非线性部分特性未知的系统，可使用线性控制器的设计方法与神经网络结合，这种控制系统有效扩展线性控制方法的实际应用，可处理原有线性控制器不能解决的不确定性非线性问题，而对于部分非线性特性已知而部分非线性特性未知的系统，将非线性控制方法（如微分几何法、动态逆方法）与神经网络结合，采用神经网络对消逆误差，可以实现系统输出特性跟踪参考模型等等。

对于控制学科而言，神经网络的巨大吸引力主要在于^[59]：神经网络本质上是

非线性系统，能够充分逼近任意复杂的非线性系统；神经网络具有高度的自适应性和自组织性，能够学习和适应严重不确定性系统的动态特性；系统信息等势分布存储在网络的各神经元及其连接权中，故有很强的鲁棒性和容错能力；信息的并行处理方式使得快速进行大量运算成为可能。这些优点说明神经网络在解决高度非线性和严重不确定性系统的控制方面具有较大的潜力。目前，基于神经网络的不确定非线性系统自适应控制和容错控制已成为控制理论研究的热点之一，引起了广大学者的关注。

1995 年，Baumgarten G 针对飞机舵面损伤提出一种基于遗传算法的自适应重构概念，将神经网络的遗传算法与模型参考自适应控制理论相结合，有效的适应了飞行系统的线性及非线性故障^[60]；1997 年，Marcello R 和 Naplitano 针对飞机作动器故障，采用在线训练的神经网络和模糊控制对非线性系统进行容错控制律设计^[61]；1990 年，Baron 等人将多项式神经网络用于故障检测和容错飞控系统设计中^[62]；W.L. Bake 和 J.A. Farrell 采用神经网络控制器设计了超机动飞行控制系统^[63]；Dong-HonShin 采用前向 BP 神经网络对非线性飞行控制系统进行容错控制，构造了神经网络自适应控制器，对飞行控制系统作动器故障进行在线重构控制，并且证明了闭环重构控制系统的稳定性^[64]。

与传统控制器相比，神经网络控制器既有优点，又有其不足。神经网络控制器最大的优势在于该方法不依赖于被控对象的数学模型，具有自学习、自适应性，因此在容错控制中得到广泛的关注；同时，神经网络学习的过程也就是控制器的调整过程，因此这种策略对网络的学习速度要求较高。

从神经网络的函数逼近功能这方面划分，神经网络分为全局逼近网络和局部逼近网络。如果神经网络的一个或者多个可调参数（比如连接权值和阈值）在输入样本空间的每一点处对任何一个输出都有影响，则称这种神经网络为全局逼近网络。上一章中介绍的 BP 神经网络是全局逼近网络的典型例子，对于每个输入/输出样本对，网络的每一个连接权都需要进行训练调整，这样做固然考虑了输入样本中的每一个输入，通过连接权的调整来实现对输出的影响；但是对每个连接权都要调整导致学习速度慢，很难满足要求在线控制的实时性要求。相对的，如果对网络输入空间的某个局部区域只有少数连接权影响网络输出，则称这种神经网络为局部逼近网络。局部逼近网络不必对所有的连接权进行训练，只需要对影响网络输出的输入连接权进行训练，因此网络学习速度快，可有效的应用于实时控制，其中 CMAC 神经网络是重要的一种局部逼近网络^[66, 67]，这是一种模拟小脑功能的局部逼近神经网络，采用查表技术，拥有计算次数少、学习速度快、局部泛化能力强、可以硬件固化等优点，非常适用于计算量大，要求学习速度快的控制领域。

尽管神经网络控制已经取得了一些令人瞩目的成就,但是神经网络控制在理论与实践上,仍然存在一些问题,如神经网络的稳定性和收敛性问题,神经网络控制的实时性问题,引入神经网络控制的系统的稳定性和收敛性的系统分析方法问题等等,这些问题还有待于进一步研究与探讨。

考虑到这些问题,对于不确定非线性的故障系统,本章提出一种 CAMC 神经网络自适应容错飞行控制方法, CAMC 学习速度快,适用于实时控制;在此基础上结合基于李亚普诺夫函数的反步递推 (backstepping) 方法设计鲁棒自适应容错控制律和神经网络权值调整算法。反步递推法比较适合在线控制,并可以保证系统的输出特性以及神经网络的学习收敛性, CMAC 同时学习故障后的不确定性和虚拟控制量导数信息,避免了 Backstepping 的计算膨胀问题,最后将该控制策略应用于某飞机的飞行仿真中进行验证。

6.2 小脑模型神经网络

小脑模型神经网络 (Cerebellar Model Articulation Controller, 简称 CMAC 网络) 由 J.S.Albus 于 1975 年提出^[68], 因仿照小脑控制肢体运动的原理而得名, 小脑指挥运动时具有不假思索就能做出条件反射、迅速响应的特点, CMAC 神经网络就是一种模拟条件反射做出快速联想的网络, 具有以下特点:

(1) 网络具有局部联想功能, 即泛化能力, 相似的输入产生相似的输出, 独立的输入产生独立的输出;

(2) 网络中只有少数神经元的连接权影响输出, 且对输出产生影响的神经元由输入决定, 输入样本决定了哪些神经元影响输出, 哪些神经元不影响;

(3) 对于每个神经元而言, 输入输出是一种线性关系, 而网络总体上是一种非线性映射的表格系统。

CMAC 是一种前馈神经网络, 由两个基本映射 (概念映射、实际映射) 表示输入输出之间的非线性关系, 是一种表达复杂非线性函数的表格查询型自适应神经网络, 通过学习算法改变表格的内容, 具有信息分类存储的能力。CMAC 中每个状态变量被量化并且问题空间被化分成离散状态。量化输入构成的向量指定了一个离散状态并且被用于产生地址来激活联想单元中存储的联想强度从而恢复这个状态的信息。CMAC 的基本结构如图 6-1 所示:

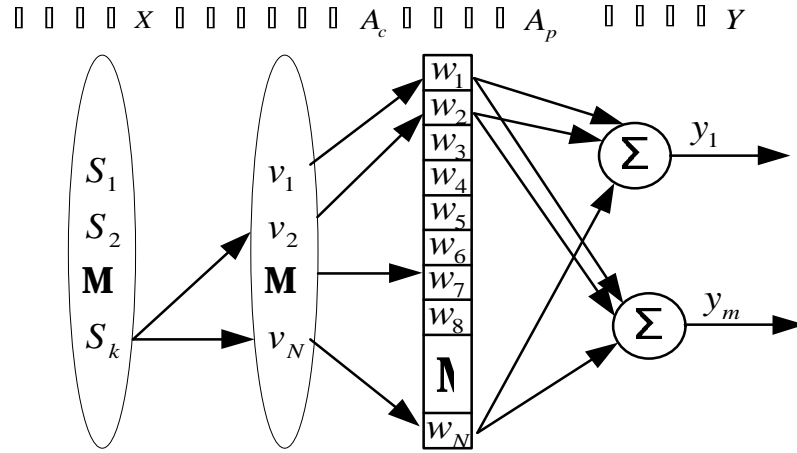


图 6-1 CMAC 神经网络结构图

CMAC 神经网络的结构分为三大块：

- (1) X ：输入空间；
- (2) A ：概念空间，包括概念存储区 A_c 和实际存储区 A_p ；
- (3) Y ：输出空间

CMAC 网络通过两个映射，实现输入输出间的非线性关系：

- (1) 概念映射 $S: X \rightarrow A, v = S(X)$

概念映射 (Conceptual Mapping) 是从输入空间 X 到概念存储空间 A_c 的映射，由 CMAC 网络的输入层来实现，具有局部泛化能力。 $v = [v_1 \ v_2 \ \mathbf{L} \ v_N]^T$ 是 N 维概念空间 A_c 中的向量， v_i 的值取 1 或 0 两个值。对于一个特定的 X ， v 向量中只有 c 个少数元素为 1，称之为激活单元；大部分元素为零。

$v = S(X)$ 实现的是一个特定的非线性映射。输入空间的一个点对应于 v 中的 c 个元素 1，即对应 A_c 中的一个局部区域。

- (2) 实际映射 $P: A \rightarrow Y, Y = L(v)$

实际映射 (Practical Mapping) 是由概念存储空间 A_c 中 c 个激活单元映射到实际存储区 A_p 中的 c 个单元，这些单元中存放着相应的网络权值，从而得到网络输出，可表示为：

$$w = [w_1 \ w_2 \ \mathbf{L} \ w_N]^T, \quad v = [v_1 \ v_2 \ \mathbf{L} \ v_N]^T$$

则有

$$y = H(x) = L(v) = \sum_{i=1}^N w_i v_i = w^T v \quad (6-1)$$

其中 v 向量中只有 c 个少数元素为 1, c 称为泛化系数, 反映了系统泛化能力的大小; w 为可调权值向量。 $Y = L(v)$ 实现的是一个线性映射, CMAC 经过两次映射, 相似的输入产生相似的输出, 独立的输入产生独立的输出。

设 y_d 为期望输出, 对应的误差表达式为:

$$E = y_d - y = y_d - \sum_{i=1}^N w_i v_i \quad (6-2)$$

CMAC 神经网络根据期望输出与实际输出的误差大小来调整权值, 在常规 CMAC 算法中, 误差被平均分配到所有激活的存储单元, 神经网络的连接权值学习算法为:

$$w_i(k+1) = w_i(k) + \frac{h(y_d - \sum_{i=1}^N w_i v_i) v_i}{c} \quad (6-3)$$

式中, $w_i(k)$ 表示第 k 次迭代后在第 i 个存储单元的权值, c 为泛化系数, h 为学习速率。可见 CMAC 网络只训练线性映射部分, 因此收敛速度快。

[定理 6.1]^[69]: (CMAC 一般估值定理) 在一紧集 $U \in R^n$ 上, 给定任意实值连续函数 $g(u)$ 及任意 $\epsilon > 0$, 则存在一个 CMAC 神经网络 $h(u)$, 使得

$$\sup |h(u) - g(u)| < \epsilon, \forall u \in U$$

式中, $u = [u_1 \ \dots \ u_N]^T$ 为输入向量。

[引理 6.2]: CMAC 神经网络采用权值调整式 (6-3), 当学习速率 $0 < h < 2$ 时, CMAC 神经网络的学习过程是收敛的。

证明: 假设理想 CMAC 的权值为 w , $w \in R^{N \times 1}$, 由式 (6-1) 可知, 理想网络的输出为:

$$y_k = (w(k))^T v(k) \quad (6-4)$$

当前 CMAC 的权值为 \hat{w} , $\hat{w} \in R^{N \times 1}$, 初始值为零向量, 则当前网络的输出为:

$$\hat{y}_k = (\hat{w}(k))^T v(k) \quad (6-5)$$

对比理想 CMAC，当前网络第 k 时刻的学习输出误差为：

$$e(k) = y_k - \hat{y}_k = (w(k))^T v(k) - (\hat{w}(k))^T v(k) \quad (6-6)$$

定义 $\mathcal{W}(k) = w(k) - (\hat{w}(k))$ 为权值误差向量，则式 (6-6) 可写为：

$$e(k) = (\mathcal{W}(k))^T v(k) \quad (6-7)$$

定义 CMAC 的 Lyapunov 函数：

$$V(k) = \mathcal{W}(k)^T \mathcal{W}(k) \quad (6-8)$$

由 CMAC 的权值调整式 (6-3) 知，

$$\mathcal{W}(k+1) - \mathcal{W}(k) = \frac{h(y_d - \sum_{i=1}^N \hat{w}_i v_i) v}{c} = \frac{he(k)v(k)}{c} \quad (6-9)$$

由式 (6-8)、(6-9) 可得：

$$\begin{aligned} V(k+1) - V(k) &= \mathcal{W}(k+1)^T \mathcal{W}(k+1) - \mathcal{W}(k)^T \mathcal{W}(k) \\ &= (\mathcal{W}(k) - \frac{he(k)v(k)}{c})^T (\mathcal{W}(k) - \frac{he(k)v(k)}{c}) - \mathcal{W}(k)^T \mathcal{W}(k) \\ &= -\frac{he^2(k)}{c} \times (2-h) \end{aligned} \quad (6-10)$$

当下式成立时，

$$V(k+1) - V(k) \leq 0, \forall k \quad (6-11)$$

GCMAC 稳定收敛。只需 $h(h-2)e^2(k)/c < 0$ ，即 $0 < h < 2$ 时，CMAC 神经网络学习收敛。CMAC 工作原理如下图所示：

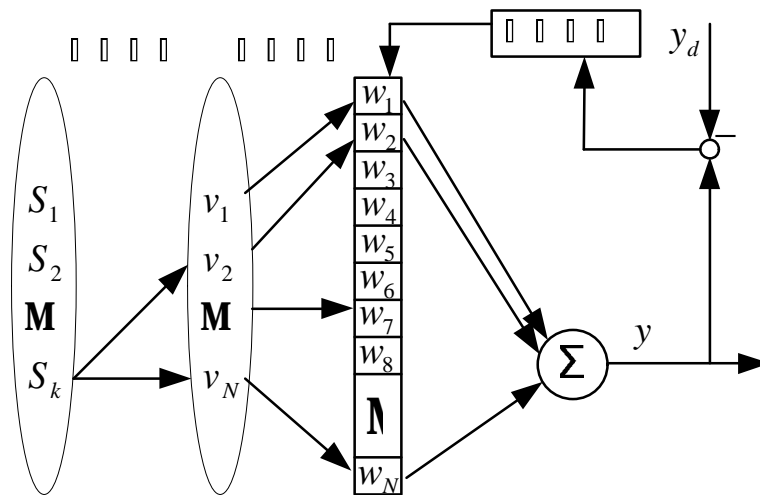


图 6-2 CMAC 结构原理

6.3 模型参考自适应逆神经网络容错控制系统

6.3.1 模型参考自适应逆控制

自适应逆控制是美国斯坦福大学著名教授 B. Widrow 于 1986 年首次提出来的，在控制系统和调节器设计中是一种很新颖的方法。该方法采用被控对象传递函数的逆做为串联控制器来对系统的动态特性作开环控制。控制器是自适应的，并将调节到使对象及其控制器的总体动态响应达到最优。

与传统的控制截然不同，反馈仅在自适应过程本身采用，用于控制系统中的可变参数，并不控制系统中信号的流动，避免了因反馈可能引起的不稳定问题，从而在动态控制性能上得到改善。

自适应逆控制的基本原理如下图 6-3 所示。该系统的目的就是要使得对象的输出跟随指令输入。由于对象一般是未知的，这就要自适应的调节控制器的参数以得到一个真正的对象的逆。按照某种自适应算法，用对象输出和指令输入之差的误差信号来调节控制器的参数以使得误差信号的均方误差最小^[70]。自适应逆控制可以不必知道系统的具体信息，这为不确定系统的控制问题提供了有效地解决途径。

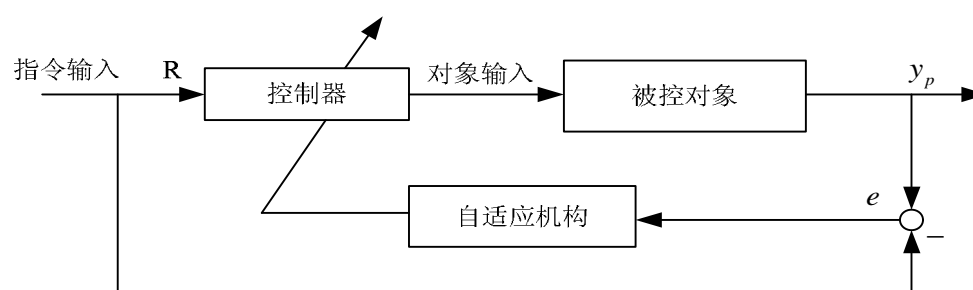


图 6-3 自适应逆控制

在飞行控制设计中，有时要求的不是对象跟踪指令输入本身，而是要跟踪一个平滑特性（参考模型），如图 6-4 所示，称为模型参考自适应逆控制系统。参考模型具有设计者对整个系统所要求的动态响应，通过自适应算法调整控制器，最终使得参考模型输出和对象输出之间的均方误差最小。此时控制器和被控对象的级联在收敛后将有类似参考模型特性的动态响应，控制器和对象传递函数的乘积近似参考模型的传递函数。

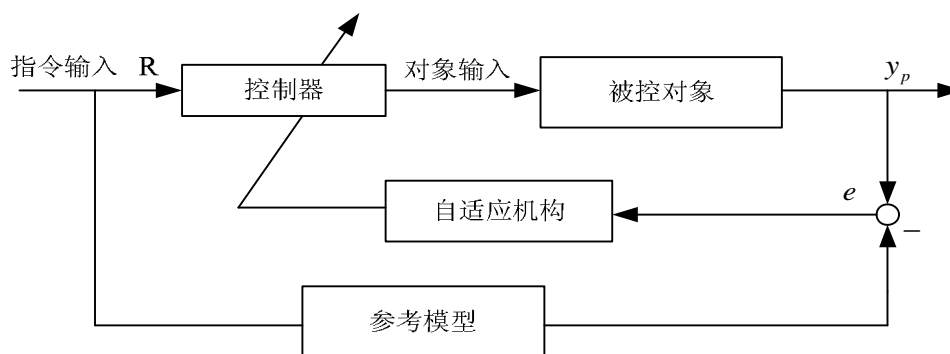


图 6-4 模型参考自适应逆控制

6.3.2 基于模型参考自适应逆的神经网络容错控制设计

基于模型参考自适应逆的神经网络容错控制是模型参考自适应逆控制理论和神经网络方法的有效结合，考虑到神经网络具有高度的自学习和自适应能力，该策略中自适应控制器由 CMAC 神经网络实现，根据误差在线调整神经网络控制器，从而实现系统输出对参考模型的跟踪。

神经网络自适应逆控制器的设计有两种方式：一种是直接法，按照期望输出与实际输出之差调节控制器，直接进行控制；另外一种间接法，通过系统辨识获得被控对象的数学模型，然后按照期望输出与实际输出之差调节控制器（如图 6-5 所示）。

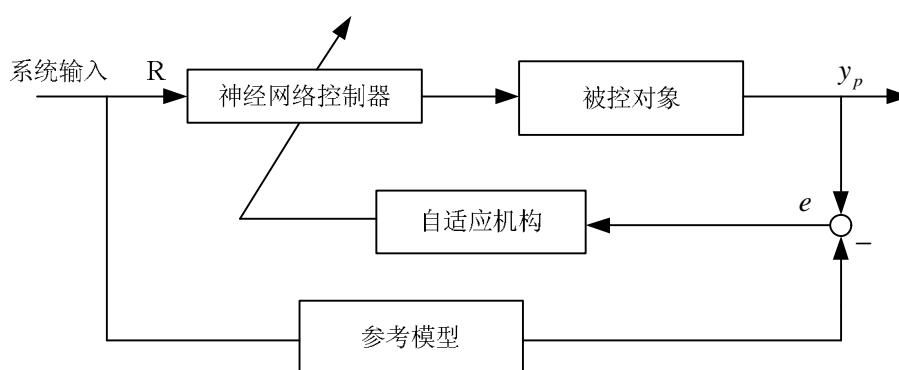


图 6-5-1 直接法设计模型参考自适应逆神经控制系统

直接法，又称为特定学习，是一种边控制边学习的设计方法，该方法把控制器设计与控制满意程度紧密结合，神经网络权值的调整依靠期望输出与被控对象实际输出之差，有助于自适应对象与环境的变化。间接法，又称为泛化学习。相比直接法，间接法引入了神经网络辨识器，利用被控对象的输入输出样本对进行在线或离线训练以逼近对象模型，为控制器提供误差，神经网络控制器自适应地学习被控对象的逆模型。辨识器的训练与控制器的训练分开进行，最终自适应控

制器产生合适的控制信号使得被控对象的输出逼近期望输出。

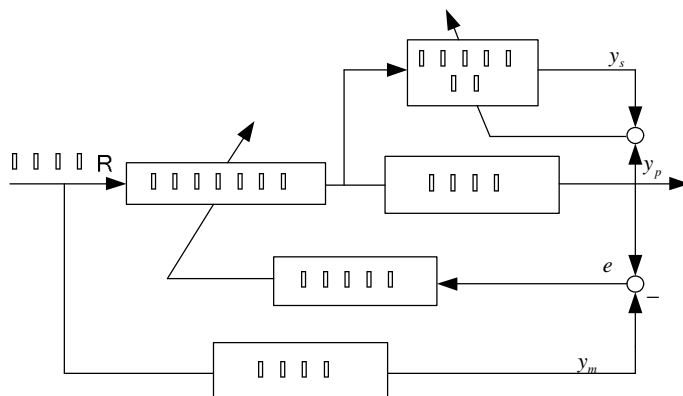


图 6-5-2 间接法设计模型参考自适应逆神经控制系统

采用间接法设计的模型参考自适应逆神经控制存在一个问题，由于训练阶段不能保证被控对象的输出在允许的范围内，导致在控制阶段对象输出可能超出允许的工作范围，使得控制失误。如果被控对象的动态特性变化明显，那么不适合使用间接设计法^[66]。考虑到这个因素以及学习效率问题，本章采用直接法设计模型参考自适应逆神经网络容错控制系统。

模型参考自适应逆神经网络容错控制理论为重构控制系统的研究提供了很好的思路。根据设计要求选择合适的参考模型，飞机发生故障时，根据参考模型输出和对象实际输出的误差，自适应的调节控制器，使得被控对象跟踪参考模型的动态响应。该容错控制策略并不需要知道故障的大小、位置以及故障特性等具体信息，通过自适应机构，可以使得飞机在故障情况下仍能满足一定的动态性能，完成重构控制系统设计。容错控制系统如下图所示：

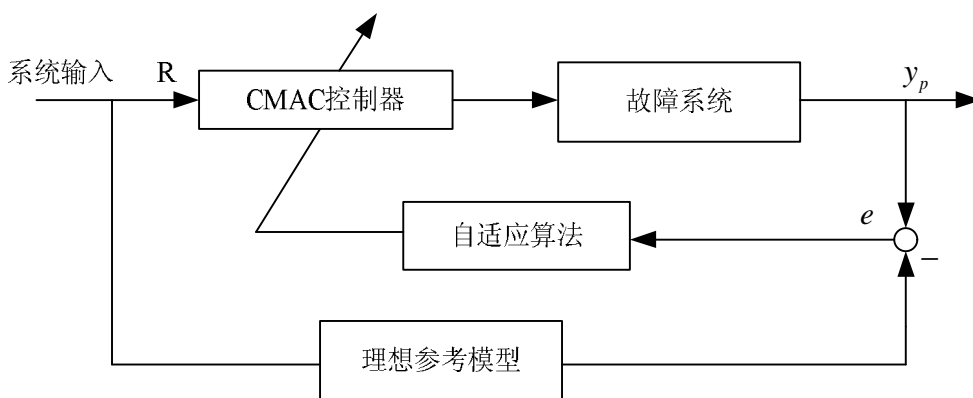


图 6-6 模型参考自适应逆神经网络容错控制系统

6.4 反步递推自适应算法

对于线性系统而言，被控对象一般都有极点和零点，因此对象的逆也应该有

零点和极点。如果被控对象是最小相位系统，即它的全部零点都在 Z 平面的单位圆内，那么逆的极点都在单位圆内，逆是稳定的。如果被控对象是非最小相位的，那么它的逆有可能某些极点在 Z 平面的单位圆外，可能存在逆是不稳定的情况。一般来说，在不知道被控对象是否是最小相位的情况下，关于完成一个对象逆的可行性上就存在着不确定性。同时由于故障可能引起的系统不确定性和非线性，导致稳定性更难以保证。前文提到的梯度下降法虽然可以调整 CMAC 控制器，但不能确切保证系统稳定。考虑到这些因素，本文采用基于 Lyapunov 稳定性的 Backstepping 反步递推自适应算法设计自适应容错控制律和神经网络权值调整算法。

Backstepping 设计方法（反步设计法或后推设计法）自 Kokotovic^[71]等人于 1991 年提出后就受到了广大学者的重视^[72-77]。Backstepping 算法是在非线性标准型下进行的，通常与李亚普诺夫型自适应控制结合使用。Backstepping 算法是交叉选择李亚普诺夫函数与反馈控制的递归过程，将系统的设计问题分解为一系列低阶子系统的设计问题，通过设计虚拟控制量由前向后递推，最终实现系统镇定或输出跟踪。

Backstepping 算法为控制李亚普诺夫函数（control Lyapunov Functions）的构造提供了一种系统化、结构化方法，在处理系统非线性和反馈控制的形式选择上具有很大的自由度，因而在非线性不确定系统控制器设计中得到了广泛应用。将反步法与 Lyapunov 型自适应控制律相结合构成自适应反步算法，可以解决非匹配不确定系统的自适应鲁棒跟踪问题。这种方法综合考虑控制律和自适应律，从而使整个闭环系统满足期望的动静态性能。在过去的十年以来，Backstepping 方法已经成为非线性控制技术发展中一个重要的组成因素，无论在理论上还是实际应用中，Backstepping 方法的研究取得了一系列有意义的成果^[78,79]。

6.4.1 问题描述

以二维系统为例，考虑如下非线性系统：

$$\begin{cases} \dot{\mathbf{x}}_1 = \mathbf{x}_2 + \mathbf{j}_1(\mathbf{x}_1)\mathbf{J} \\ \dot{\mathbf{x}}_2 = \mathbf{u} + \mathbf{j}_2(\mathbf{x}_2)\mathbf{J} \end{cases} \quad (6-12)$$

式中， $\mathbf{x} = [\mathbf{x}_1 \quad \mathbf{x}_2]^T \in R^2$ 为系统状态变量， $\mathbf{u} \in R$ 为系统控制量， $\mathbf{j}_i(\mathbf{x}_i, \mathbf{L} \mathbf{x}_i)$ 为非线性函数。

Backstepping 算法的基本设计思想是：把每一个子系统 $\dot{\mathbf{x}}_i = \mathbf{x}_{i+1} + \mathbf{j}_i(\mathbf{x}_i)\mathbf{J}_i$ 中的 \mathbf{x}_{i+1} 作为虚拟控制输入，通过设计适当的虚拟反馈 $\mathbf{x}_{i+1} = \mathbf{a}_i(\mathbf{x}_i, \mathbf{J}_i)$ ，使得前面的子

系统得到镇定，状态达到渐近稳定。由于系统的虚拟反馈 $\mathbf{a}_i(x_i, \mathbf{J}_i)$ 一般不可得，因此引入误差变量 $z_{i+1} = x_{i+1} - \mathbf{a}_i(x_i, \mathbf{J}_i)$ ，对每一步构造一个 Lyapunov 函数，通过控制输入的作用，使得 x_{i+1} 渐近收敛于 $\mathbf{a}_i(x_i, \mathbf{J}_i)$ ，每一步的虚拟误差状态分量 z_i 渐近收敛于零，最终实现整个系统的渐近稳定，其设计步骤如下：

第一步，将 x_2 看作子系统的控制输入，设计自适应控制器^[80]：

$$x_2 = \mathbf{a}_1(x_1, \mathbf{J}_1) = c_1 z_1 - \mathbf{j}_1(x_1) \mathbf{J}_1 \quad (6-13)$$

$$\dot{\mathbf{J}}_1 = g \mathbf{z} \mathbf{j}_1(x_1) \quad (6-14)$$

上式中 $z_1 = x_1$ ， \mathbf{J}_1 为第一步中参数 \mathbf{J} 的估计， c_1 为正实数。

由式 (6-13) 定义误差变量 z_2 ：

$$z_2 = x_2 - \mathbf{a}_1(x_1, \mathbf{J}_1) = -c_1 z_1 + \mathbf{j}_1(x_1) \mathbf{J}_1 + x_2 \quad (6-15)$$

则有：

$$\dot{z}_1 = -c_1 z_1 + \mathbf{j}_1(x_1)(\mathbf{J} - \mathbf{J}_1) + z_2 \quad (6-16)$$

定义 Lyapunov 函数如下：

$$V_1(z_1, \mathbf{J}_1) = \frac{1}{2} z_1^2 + \frac{1}{2g} (\mathbf{J} - \mathbf{J}_1)^2 \quad (6-17)$$

对 Lyapunov 函数 V_1 求导，可得：

$$\dot{V}_1 = z_1 \dot{z}_1 - \frac{1}{g} (\mathbf{J} - \mathbf{J}_1) \dot{\mathbf{J}} = z_1 z_2 - c_1 z_1^2 \quad (6-18)$$

第二步，对 z_2 求导，可得：

$$\dot{\mathbf{x}}_2 = \dot{\mathbf{x}}_2 - \dot{\mathbf{a}}_1 = u + \mathbf{j}_2 \mathbf{J} - \frac{\partial \mathbf{a}_1}{\partial z_1} \dot{\mathbf{x}}_1 - \frac{\partial \mathbf{a}_1}{\partial \mathbf{J}_1} \dot{\mathbf{J}}_1 \quad (6-19)$$

将式 (6-13)、(6-16) 代入上式中，则有：

$$\dot{\mathbf{x}}_2 = u - \frac{\partial \mathbf{a}_1}{\partial z_1} x_2 - \frac{\partial \mathbf{a}_1}{\partial \mathbf{J}_1} g \mathbf{j}_1 z_1 + (\mathbf{j}_2 - \frac{\partial \mathbf{a}_1}{\partial z_1} \mathbf{j}_1) \mathbf{J} \quad (6-20)$$

定义 Lyapunov 函数 $V_2(z_1, z_2, \mathbf{J}_1) = V_1 + \frac{1}{2} z_2^2$ ，并对 Lyapunov 函数 V_2 求导，可得：

$$\begin{aligned}\dot{V}_2 &= \dot{V}_1 + z_2 \dot{z}_2 \\ &= -c_1 z_1^2 + z_2 [u + z_1 - \frac{\partial a_1}{\partial z_1} x_2 - \frac{\partial a_1}{\partial J_1} g j_1 z_1 + (j_2 - \frac{\partial a_1}{\partial z_1} j_1) J_1] \end{aligned} \quad (6-21)$$

应用已存在的参数估计 J_1 设计自适应控制器如下：

$$u = -c_2 z_2 - z_1 - \frac{\partial a_1}{\partial z_1} x_2 - \frac{\partial a_1}{\partial J_1} g j_1 z_1 + (j_2 - \frac{\partial a_1}{\partial z_1} j_1) J_1 \quad (6-22)$$

则有：

$$\dot{V}_2 = -c_1 z_1^2 - c_2 z_2^2 + z_2 (j_2 - \frac{\partial a_1}{\partial z_1} j_1) (J - J_1) \quad (6-23)$$

用 J_2 代替 J_1 ，可得控制器：

$$u = -c_2 z_2 - z_1 - \frac{\partial a_1}{\partial z_1} x_2 - \frac{\partial a_1}{\partial J_1} g j_1 z_1 + (j_2 - \frac{\partial a_1}{\partial z_1} j_1) J_2 \quad (6-24)$$

此时 \dot{z}_2 可表示为：

$$\dot{z}_2 = -c_1 z_1^2 - c_2 z_2^2 + z_2 (j_2 - \frac{\partial a_1}{\partial z_1} j_1) (J - J_2) \quad (6-25)$$

考虑到参数 J_2 ，对于整个系统定义 Lyapunov 函数 V_2 ：

$$V_2(z_1, z_2, J_1, J_2) = V_1(z_1, J_1) + \frac{1}{2} z_2^2 - \frac{1}{2g} (J - J_2)^2 \quad (6-26)$$

对 Lyapunov 函数 $V_2(z_1, z_2, J_1, J_2)$ 求导，可得：

$$\begin{aligned}\dot{V}_2 &= \dot{V}_1 + z_2 \dot{z}_2 - \frac{1}{g} (J - J_2) \dot{J} \\ &= -c_1 z_1^2 - c_2 z_2^2 + [(j_2 - \frac{\partial a_1}{\partial z_1} j_1) z_2 - \frac{1}{g} \dot{J}_2] (J - J_2) \end{aligned} \quad (6-27)$$

令 $\dot{J}_2 = g(j_2 - \frac{\partial a_1}{\partial z_1} j_1) z_2$ ，有：

$$\dot{V}_2 = -c_1 z_1^2 - c_2 z_2^2 \quad (6-28)$$

综上所述：

$$\left\{ \begin{array}{l} \dot{J}_1 = g j_1(x_1) \\ \dot{z}_1 = -c_1 z_1 + j_1(x_1)(J - J_1) + z_2 \\ \dot{J}_2 = g(j_2 - \frac{\partial a_1}{\partial z_1} j_1) z_2 \\ \dot{z}_2 = -c_1 z_1^2 - c_2 z_2 + z_2(j_2 - \frac{\partial a_1}{\partial z_1} j_1)(J - J_2) \\ u = -c_2 z_2 - z_1 - \frac{\partial a_1}{\partial z_1} x_2 - \frac{\partial a_1}{\partial J_1} g j_1 z_1 + (j_2 - \frac{\partial a_1}{\partial z_1} j_1) J_2 \end{array} \right. \quad (6-29)$$

以上是 backstepping 算法的设计过程。由上述推导可见，Backstepping 算法是一种由前向后递推的设计方法，它将复杂的非线性系统分解成不超过系统阶数的子系统，然后为每个子系统设计部分 Lyapunov 函数和中间虚拟控制量，一直后推到整个系统，将他们集成起来完成整个控制律的设计最终实现系统的输出跟踪。

反步法中的虚拟控制是一种静态补偿思想，前一个子系统必须通过其后子系统的虚拟控制才能达到镇定目的。因此这种设计方法要求系统的结构必须是与式 (6-12) 类似的严参数反馈系统。

6.4.2 基于 backstepping 算法的神经网络容错飞行控制方法

考虑非匹配不确定非线性飞行系统：

$$\left\{ \begin{array}{l} \dot{\mathbf{x}}_1 = f_1(x_1) + g_1(x_1)x_2 + \Delta_1 \\ \dot{\mathbf{x}}_2 = f_2(x_1, x_2) + g_2(x_1, x_2)x_3 + \Delta_2 \\ \dots\dots\dots \\ \dot{\mathbf{x}}_n = f_n(x_1, \mathbf{L}, x_n) + g_n(x_1, \mathbf{L}, x_n)u + \Delta_n \\ y = x_1 \end{array} \right. \quad (6-30)$$

在上式中， $x = [x_1 \quad \mathbf{L} \quad x_n]^T \in R^T$ 为系统状态变量， $u \in R$ 为系统控制量， y 为系统输出， $f_i(x_1, \mathbf{L}, x_i)$ 、 $g_i(x_1, \mathbf{L}, x_i)$ 为非线性函数， Δ_i 表示故障系统的不确定性。控制目标是设计鲁棒自适应容错控制器，以保证故障后的系统 (6-30) 的输出渐近跟踪期望输出。

前文中提到 CMAC 神经网络是一种基于局部学习的神经网络，学习速度快，具有较好的逼近性能，非常适合于实时控制。CMAC 的函数逼近可以表示为：

$$y = H(x) = L(v) = \sum_{i=1}^N w_i v_i = w^T v$$

其中, w 为可调权值向量; $v = [v_1 \ v_2 \ \mathbf{L} \ v_N]^T$, v 向量中只有 c 个少数元素为 1, 其余为 0。

且有[定理 6.1]: (CMAC 一般估值定理) 在一紧集 $U \in R^n$ 上, 给定任意实值连续函数 $g(u)$ 及任意 $\epsilon > 0$, 则存在一个 CMAC 神经网络 $h(u)$, 使得

$$\sup |h(u) - g(u)| < \epsilon, \forall u \in U$$

式中, $u = [u_1 \ \dots \ u_N]^T$ 为输入向量。

本章采用 CMAC 神经网络, 结合 backstepping 算法设计了鲁棒自适应控制器, 具体步骤如下。

Step 1, 定义虚拟反馈误差 $z_1 = x_1 - x_{1d} = x_1 - y_d$, $z_2 = x_2 - x_{2d} = x_2 - a_1$, 其中, x_{1d} 为期望状态变量, $x_{1d} = y_d$, 其中 y_d 可微, 为系统期望输出。

对 z_1 求导, 可得:

$$\dot{\mathbf{x}} = f_1(x_1) + g_1(x_1)z_2 + g_1(x_1)a_1 + \Delta_1 - \dot{\mathbf{x}}_d \quad (6-31)$$

由于 (6-31) 式中不确定项 Δ_1 不可获得, 因此我们采用 CMAC 神经网络在线学习逼近, 即

$$\Delta_1 = \mathbf{w}_1^T \mathbf{v}_1 = (\hat{\mathbf{w}}_1^T + \mathbf{w}_1^T) \mathbf{v}_1 \quad (6-32)$$

式中, \mathbf{w}_1 为 CMAC 理想权值向量, $\hat{\mathbf{w}}_1$ 为 CMAC 权向量的估计, \mathbf{w}_1 为 CMAC 误差权向量。

定义虚拟控制输入为:

$$a_1 = g_1^{-1}(x_1) \left(-f_1(x_1) - z_1 - \hat{\mathbf{w}}_1^T \mathbf{v}_1 + \dot{\mathbf{x}}_d \right) \quad (6-33)$$

将式 (6-32)、(6-33) 代入式 (6-31), 可得:

$$\dot{\mathbf{x}} = -z_1 + g_1(x_1)z_2 + \mathbf{w}_1^T \mathbf{v}_1 \quad (6-34)$$

定义 Lyapunov 函数为 $V_1 = \frac{1}{2} z_1^2 + \frac{1}{2b_1} \mathbf{w}_1^T \mathbf{w}_1$, 对其求导可得:

$$\begin{aligned}\dot{v}_1 &= z_1 \dot{z}_1 + \frac{1}{b_1} \dot{w}_1^T w_1 \\ &= -z_1^2 + g_1(x_1) z_1 z_2 + \dot{w}_1^T (v_1 z_1 + \frac{1}{b_1} w_1)\end{aligned}\quad (6-35)$$

选择 CMAC 神经网络的权值学习规则为：

$$\dot{w}_1 = -b_1 v_1 z_1 \quad (6-36)$$

其中， $0 < c_1 b_1 < 2$ ， c_1 为神经网络的泛化系数。

把式 (6-36) 代入式 (6-35)，可得：

$$\dot{v}_1 = -z_1^2 + g_1(x_1) z_1 z_2 \quad (6-37)$$

Step 2，定义虚拟反馈误差 $z_3 = x_3 - a_2$ ，对 z_2 求导可得：

$$\dot{z}_2 = \dot{z}_2 - \dot{a}_1 = f_2(x_1, x_2) + g_2(x_1, x_2) z_3 + g_2(x_1, x_2) a_2 + \Delta_2 - \dot{a}_1 \quad (6-38)$$

由于 (6-38) 式中不确定项 Δ_i 不可获得，因此我们采用 CMAC 神经网络在线学习逼近，即

$$\Delta_i = w_i^T v_i = (\hat{w}_i^T + \dot{w}_i^T) v_i \quad (6-39)$$

式中， w_i 为 CMAC 理想权值向量， \hat{w}_i 为 CMAC 权向量的估计， \dot{w}_i 为 CMAC 误差权向量。

定义虚拟控制输入为：

$$a_2 = g_2^{-1} \left(-z_2 - f_2 - g_1 z_1 - \hat{w}_2^T v_2 + \dot{a}_1 \right) \quad (6-40)$$

将式 (6-39)、(6-40) 代入式 (6-38)，可得

$$\dot{z}_2 = -g_1 z_1 - z_2 + g_2 z_3 + \dot{w}_2^T v_2 \quad (6-41)$$

定义 Lyapunov 函数为 $V_2 = V_1 + \frac{1}{2} z_2^2 + \frac{1}{2b_2} \dot{w}_2^T \dot{w}_2$ ，对其求导可得：

$$\dot{V}_2 = -z_1^2 - z_2^2 + g_2 z_2 z_3 + \dot{w}_2^T v_2 z_2 + \frac{1}{b_2} \dot{w}_2^T \dot{w}_2 \quad (6-42)$$

选择 CMAC 神经网络的权值学习规则为：

$$\dot{w}_2 = -b_2 v_2 z_2 \quad (6-43)$$

其中, $0 < c_2 b_2 < 2$, c_2 为神经网络的泛化系数。

把式 (6-43) 代入式 (6-42), 可得:

$$\dot{V}_2 = -z_1^2 - z_2^2 + g_2 z_2 z_3 \quad (6-44)$$

Step i , 定义虚拟反馈误差 $z_{i+1} = x_{i+1} - a_i$, 对 z_i 求导可得:

$$\dot{\mathbf{z}}_i = \dot{\mathbf{z}}_i - \dot{\mathbf{a}}_{i-1} = f_i + g_i z_{i+1} + g_i a_i + \Delta_i - \dot{\mathbf{a}}_{i-1} \quad (6-45)$$

同样, 设计 CMAC 神经网络在线逼近系统不确定性, 即:

$$\Delta_i = \mathbf{w}_i^T \mathbf{v}_i = (\hat{\mathbf{w}}_i^T + \mathbf{w}_i^T) \mathbf{v}_i \quad (6-46)$$

式中, \mathbf{w}_i 为 CMAC 理想权值向量, $\hat{\mathbf{w}}_i$ 为 CMAC 权向量的估计, \mathbf{w}_i^T 为 CMAC 误差权向量。

定义虚拟控制输入为:

$$\mathbf{a}_i = g_i^{-1} (-z_i - f_i - g_{i-1} z_{i-1} - \hat{\mathbf{w}}_i^T \mathbf{v}_i + \dot{\mathbf{a}}_{i-1}) \quad (6-47)$$

将式 (6-46)、(6-47) 代入式 (6-45), 可得:

$$\dot{\mathbf{z}}_i = -g_{i-1} z_{i-1} - z_i + g_i z_{i+1} + \mathbf{w}_i^T \mathbf{v}_i \quad (6-48)$$

定义 Lyapunov 函数为 $V_i = V_{i-1} + \frac{1}{2} z_i^2 + \frac{1}{2b_i} \mathbf{w}_i^T \mathbf{w}_i$, 对其求导可得:

$$\dot{V}_i = -\sum_{j=1}^i z_j^2 + g_i z_i z_{i+1} + \mathbf{w}_i^T \mathbf{v}_i z_i + \frac{1}{b_i} \mathbf{w}_i^T \dot{\mathbf{w}}_i \quad (6-49)$$

选择 CMAC 神经网络的权值学习规则为:

$$\dot{\mathbf{w}}_i = -b_i \mathbf{v}_i z_i \quad (6-50)$$

其中, $0 < c_i b_i < 2$, c_i 为神经网络的泛化系数。

把式 (6-50) 代入式 (6-49), 可得

$$\dot{V}_i = -\sum_{j=1}^i z_j^2 + g_i z_i z_{i+1} \quad (6-51)$$

第 n 步, 同样对 z_n 求导, 可有:

$$\begin{aligned} \dot{\mathbf{z}}_n &= \dot{\mathbf{z}}_n - \dot{\mathbf{a}}_{n-1} \\ &= f_n(x_1, \mathbf{L}, x_n) + g_n(x_1, \mathbf{L}, x_n)u + \Delta_n(x, u) - \dot{\mathbf{a}}_{n-1} \end{aligned} \quad (6-52)$$

采用 CMAC 神经网络在线学习逼近系统不确定性，即

$$\Delta_i = \mathbf{w}_i^T \mathbf{v}_i = (\hat{\mathbf{w}}_i^T + \mathbf{\hat{w}}_i^T) \mathbf{v}_i \quad (6-53)$$

式中， \mathbf{w}_i 为 CMAC 理想权值向量， $\hat{\mathbf{w}}_i$ 为 CMAC 权向量的估计， $\mathbf{\hat{w}}_i^T$ 为 CMAC 误差权向量。

取系统控制输入为：

$$u = g_n^{-1} \left(-z_n - f_n - \hat{\mathbf{w}}_n^T \mathbf{v}_n - g_{n-1} z_{n-1} + \dot{\mathbf{a}}_{n-1} \right) \quad (6-54)$$

定义 Lyapunov 函数 $V_n = V_{n-1} + \frac{1}{2} z_n^2 + \frac{1}{2b_n} \mathbf{\hat{w}}_n^T \mathbf{\hat{w}}_n$ ，并且选择 CMAC 神经网络权值学习规则为：

$$\dot{\mathbf{\hat{w}}}_n = -b_n \mathbf{v}_n z_n \quad (6-55)$$

其中， $0 < c_n b_n < 2$ ， c_n 为神经网络的泛化系数。

则有：

$$\dot{V}_n = -\sum_{j=1}^n z_j^2 \quad (6-56)$$

由以上分析可得， $V_n = \frac{1}{2} \sum_{j=1}^n z_j^2 + \frac{1}{2} \sum_{j=1}^n \frac{\mathbf{\hat{w}}_j^T \mathbf{\hat{w}}_j}{b_j}$ 即是整个系统的 Lyapunov 函数，其导数

$\dot{V}_n < 0$ 。由李亚普诺夫稳定性定理可知，系统是稳定的，所有状态跟踪误差以及 CMAC 神经网络权值误差渐近收敛于零。

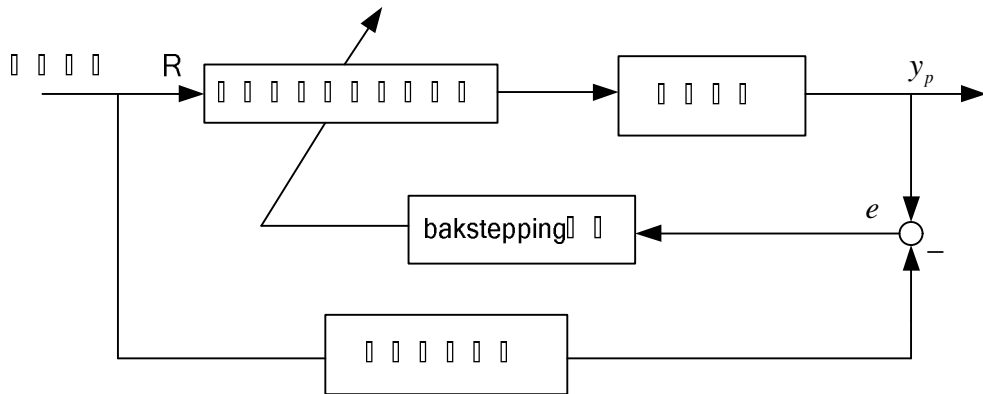


图 6-7 基于 Backstepping 算法的神经网络容错控制系统

由于反步设计算法在递推计算中需要对虚拟控制量求导，因此当系统阶数较

高时会导致较高阶虚拟控制律和最终控制器所含项随着系统阶数的增加而爆炸性膨胀，导致控制器变得复杂。为此，下面采用 CMAC 神经网络在线学习系统不确定性以及各阶虚拟控制量的导数信息，从而避免了系统阶次较高时引起的计算膨胀问题，具体算法如下。

对于 backstepping 算法的第 i 步，可改进为：

定义虚拟反馈误差 $z_{i+1} = x_{i+1} - a_i$ ，对 z_i 求导可得：

$$\dot{z}_i = \dot{x}_i - \dot{a}_{i-1} = f_i + g_i z_{i+1} + g_i a_i + \Delta_i - \dot{a}_{i-1} \quad (6-57)$$

在本步，设计 CMAC 神经网络在线学习系统不确定性和虚拟控制量导数信息，即：

$$\Delta_i - \dot{a}_{i-1} = w_i^T v_i = (\hat{w}_i^T + \tilde{w}_i^T) v_i \quad (6-58)$$

式中， w_i 为 CMAC 理想权值向量， \hat{w}_i 为 CMAC 权向量的估计， \tilde{w}_i 为 CMAC 误差权向量。

定义虚拟控制输入为：

$$a_i = g_i^{-1} (-z_i - f_i - g_{i-1} z_{i-1} - \hat{w}_i^T v_i) \quad (6-59)$$

将式 (6-58)、(6-59) 代入式 (6-57)，可得：

$$\dot{z}_i = -g_{i-1} z_{i-1} - z_i + g_i z_{i+1} + \tilde{w}_i^T v_i \quad (6-60)$$

定义 Lyapunov 函数为 $V_i = V_{i-1} + \frac{1}{2} z_i^2 + \frac{1}{2b_i} \tilde{w}_i^T \tilde{w}_i$ ，对其求导可得：

$$\dot{V}_i = -\sum_{j=1}^i z_j^2 + g_i z_i z_{i+1} + \tilde{w}_i^T v_i z_i + \frac{1}{b_i} \tilde{w}_i^T \dot{\tilde{w}}_i \quad (6-61)$$

选择 CMAC 神经网络的权值学习规则为：

$$\dot{\tilde{w}}_i = -b_i v_i z_i \quad (6-62)$$

其中， $0 < c_i b_i < 2$ ， c_i 为神经网络的泛化系数。

把式 (6-62) 代入式 (6-61)，可得

$$\dot{V}_i = -\sum_{j=1}^i z_j^2 + g_i z_i z_{i+1} \quad (6-63)$$

6.5 仿真算例

某型飞机纵向微分方程可描述为：

$$\begin{cases} \dot{\alpha} = -\frac{qsc_y^a}{mv}\alpha + \frac{g}{v}\cos(J-\alpha) + w_z - \frac{qsc_y^{d_z}}{mv}d_z \\ \dot{J} = w_z \\ \dot{w}_z = M_z^{\alpha}\alpha + M_z^w w_z + M_z^{d_z}d_z \end{cases} \quad (6-64)$$

式中， J, α, w_z 分别为俯仰角、攻角、俯仰角速度； c_y^a 为气动参数， $M_z^{\alpha}, M_z^w, M_z^{d_z}$ 为纵向力矩， d_z 为升降舵偏转角度。由于升降舵偏角 d_z 对升力的影响远小于其对力矩的影响，因此，为了控制器的设计方便，我们忽略升降舵偏角 d_z 对攻角 α 的影响，将其视为不确定项，并用航迹倾斜角 $q = J - \alpha$ 代替状态变量俯仰角 J 。

定义状态变量 $x = \left[\frac{mv}{qsc_y^a}q, \alpha, w_z \right]^T$ ，控制量 $u = d_z$ ，则故障系统状态可描述

为：

$$\begin{cases} \dot{x}_1 = f_1(x_1) + x_2 + \Delta_1 \\ \dot{x}_2 = f_2(x_1, x_2) + x_3 + \Delta_2 \\ \dot{x}_3 = f_3(x_1, x_2, x_3) + M_z u + \Delta_3 \end{cases} \quad (6-65)$$

式 (6-65) 中：

$$\begin{cases} f_1(x_1) = \frac{mg}{qsc_y^a} \cos\left(\frac{qsc_y^a}{mv}x_1\right) \\ f_2(x_1, x_2) = \frac{g}{v} \cos\left(\frac{qsc_y^a}{mv}x_1\right) - \frac{qsc_y^a}{mv}x_2 \\ f_3(x_1, x_2, x_3) = M_z^{\alpha}\dot{x}_2 + M_z^w x_3 \end{cases} \quad (6-66)$$

Δ_i 为建模和故障引起的系统不确定性。

容错控制器的设计目的为使得故障系统的纵向航迹倾斜角渐近跟踪期望航迹倾斜角 q_d 。

定义状态误差：

$$e_i = x_i - x_{id}, \quad i = 1, 2, 3 \quad (6-67)$$

其中, x_{id} 为期望状态变量值, $x_{1d} = q_d$ 为系统的期望航迹倾斜角。

对 e_1 进行求导可得

$$\dot{\mathbf{e}}_1 = f_1(x_1) + e_2 + x_{2d} + \Delta_1 - \dot{\mathbf{x}}_{1d} \quad (6-68)$$

设计 CMAC 神经网络在线学习逼近系统不确定性以及虚拟控制量的导数信息, 即有:

$$\Delta_1 - \dot{\mathbf{x}}_{1d} = \mathbf{w}_i^T \mathbf{v}_i = (\hat{\mathbf{w}}_i^T + \mathbf{w}_i^T) \mathbf{v}_i \quad (6-69)$$

式中, \mathbf{w}_i 为理想权值向量, $\hat{\mathbf{w}}_i$ 为权向量的估计, \mathbf{w}_i^T 为误差权向量。

对于第一个状态, 定义 Lyapunov 函数为 $V_1 = \frac{1}{2} e_1^2 + \frac{1}{2b_1} \mathbf{w}_1^T \mathbf{w}_1$, 选取虚拟控制量:

$$\begin{aligned} x_{2d} &= -f_1(x_1) - \hat{\mathbf{w}}_1^T \mathbf{v}_1 - k_1 e_1 \\ &= -\frac{mg}{qsc_y^a} \cos\left(\frac{qsc_y^a}{mv} x_1\right) - \hat{\mathbf{w}}_1^T \mathbf{v}_1 - k_1 e_1 \end{aligned} \quad (6-70)$$

则有:

$$\begin{cases} \dot{\mathbf{e}}_1 = -k_1 e_1 + e_2 + \mathbf{w}_1^T \mathbf{v}_1 \\ \dot{\mathbf{w}}_1 = -k_1 e_1^2 + e_1 e_2 + \mathbf{w}_1^T (e_1 \mathbf{v}_1 + \frac{1}{b_1} \mathbf{w}_1) \end{cases} \quad (6-71)$$

令 CMAC 神经网络权值调整规则为: $\dot{\mathbf{w}}_1 = -b_1 e_1 \mathbf{v}_1$, $0 < c_1 b_1 < 2$, c_1 为神经网络的泛化系数, 则有:

$$\dot{\mathbf{w}}_1 = -k_1 e_1^2 + e_1 e_2 \quad (6-72)$$

同理对于第二个状态, 定义 Lyapunov 函数 $V_2 = V_1 + \frac{1}{2} e_2^2 + \frac{1}{2b_2} \mathbf{w}_2^T \mathbf{w}_2$, 选取虚拟控制量为:

$$\begin{aligned} x_{3d} &= -f_2(x_1, x_2) - \hat{\mathbf{w}}_2^T \mathbf{v}_2 - k_2 e_2 - e_1 \\ &= -\frac{g}{v} \cos\left(\frac{qsc_y^a}{mv} x_1\right) + \frac{qsc_y^a}{mv} x_2 - \hat{\mathbf{w}}_2^T \mathbf{v}_2 - k_2 e_2 - e_1 \end{aligned} \quad (6-73)$$

并令 $\dot{v}_2 = -b_2 e_2 v_2$, $0 < c_2 b_2 < 2$, c_2 为神经网络的泛化系数, 则有:

$$\begin{cases} \dot{e}_2 = -e_1 - k_2 e_2 + e_3 + \dot{v}_2^T v_2 \\ \dot{v}_2 = -k_1 e_1^2 - k_2 e_2^2 + e_2 e_3 \end{cases} \quad (6-74)$$

对于第 3 个状态, 定义 Lyapunov 函数 $V_3 = V_2 + \frac{1}{2} e_3^2 + \frac{1}{2b_3} \dot{v}_3^T \dot{v}_3$, 选取期望控制量:

$$\begin{aligned} u &= M_z^{-1} [-f_3(x_1, \mathbf{L}, x_3) - \hat{w}_3^T v_3 - k_3 e_3 - e_2] \\ &= M_z^{-1} \left[-M_z^{\#} \dot{e}_2 + M_z^w x_3 - \hat{w}_3^T v_3 - k_3 e_3 - e_2 \right] \end{aligned} \quad (6-75)$$

并令 $\dot{v}_3 = -b_3 e_3 v_3$, $0 < c_3 b_3 < 2$, c_3 为神经网络的泛化系数, 则有:

$$\begin{cases} \dot{e}_3 = -e_2 - k_3 e_3 + \dot{v}_3^T v_3 \\ \dot{v}_3 = -k_1 e_1^2 - k_2 e_2^2 - k_3 e_3^2 \end{cases} \quad (6-76)$$

由以上分析可得, $V_3 = \frac{1}{2} \sum_{j=1}^3 e_j^2 + \frac{1}{2} \sum_{j=1}^3 \frac{\dot{v}_j^T \dot{v}_j}{b_j}$ 即是整个系统的 Lyapunov 函数,

其导数 $\dot{V}_3 < 0$ 。由李亚普诺夫稳定性定理可知, 系统是稳定的, 所有状态跟踪误差以及 CMAC 神经网络权值误差渐近收敛于零。

容错控制过程中将飞机实际高度与期望高度的误差转化为期望航迹倾斜角, 对飞机姿态进行控制, 使得飞机在着陆过程中精确跟踪期望轨迹。采用的期望航迹倾斜角指令为梯形指令, 幅值变化为 5 度, 响应如下图所示:

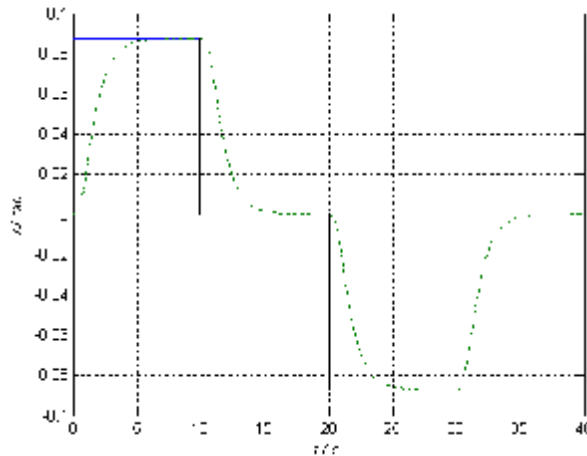


图 6-8 航迹倾斜角跟踪曲线

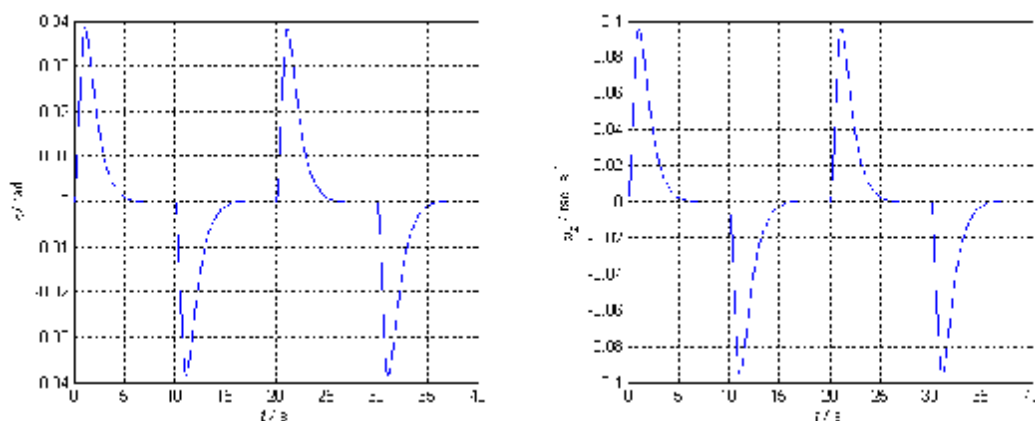
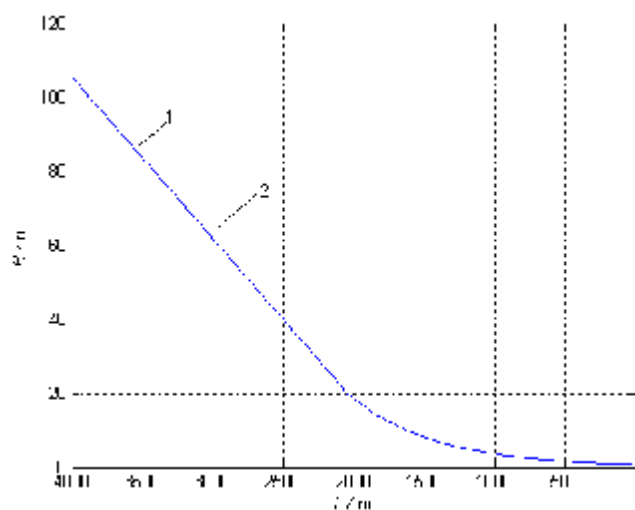


图 6-9 迎角和俯仰速率响应曲线

图 6-10 为容错控制飞机自动着陆的仿真结果。在自动着陆仿真中，主要考虑下滑段和拉平段，即将期望轨迹分为两段：下滑段采用 -2.5° 下滑角，期望轨迹为： $h_g = \text{tg}(-2.5 / 57.3) \times (l - 2042) + h_0$ ；而拉平段采用指数曲线，期望轨迹为： $h_g = (h_0 + h_B)e^{-l/t} - h_B$ ，式中， $h_0 = 20$ ， $h_B = 1.655$ 。



1-期望轨迹 2-实际轨迹

图 6-10 飞机自动着陆仿真结果

图 6-11 为着陆过程高度轨迹跟踪误差，可以看出，飞机着陆过程中下滑段和拉平段跟踪性能都很好。仅在下滑和拉平的起始阶段，由于飞机起始平飞和前一段下滑产生的惯性而使得跟踪有些滞后，达到了较高的轨迹跟踪精度，容错控制达到了理想的效果。

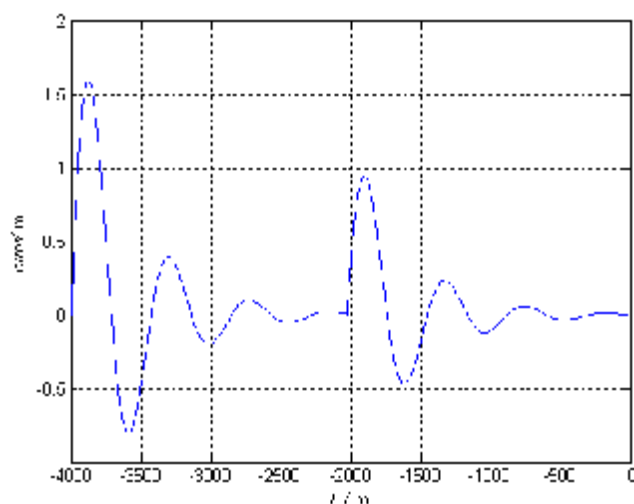


图 6-11 飞机自动着陆跟踪误差

6.6 小结

对于非线性不确定的飞行控制系统故障,本章结合局部逼近神经网络和一种系统化的设计方法-Backstepping 的思想,提出一种自适应鲁棒容错飞行控制方法。控制器采用 CMAC 神经网络,学习速度快,适用于实时控制;同时为了保证系统的稳定性和网络学习收敛,采用 Backstepping 学习算法设计虚拟控制量由前向后递推,最终实现系统镇定和输出跟踪,本策略中 CMAC 同时学习故障后的不确定性和虚拟控制量导数信息,从而避免 Backstepping 的计算膨胀问题。仿真结果表明,容错控制达到了理想的效果。

第七章 总结与展望

7.1 工作总结

飞行控制系统是现代飞机最重要的组成部分之一，它对飞机的飞行品质和安全性能起着决定性作用。由于现代飞机的系统结构日趋庞大，飞行环境相对恶劣，飞机任务复杂多变，安全可靠已成为飞行控制系统设计中必须首先考虑的问题。容错控制为提高现代飞行系统的可靠性和安全性开辟了一条新的途径，可以在系统发生故障的情况下，自动补偿故障的影响，维护系统的稳定性，并且尽可能的恢复系统正常工作状态的性能，从而保证系统的运行稳定、可靠。

本文以飞行控制系统为应用背景，研究了几种重要的容错控制方法，并进行了仿真验证，主要的研究成果和结论如下：

(1) 提出一种特征结构配置重构控制设计方法。通过对系统进行响应分析，提出一种综合了特征值灵敏度和特征向量的寻优设计指标；在此基础上结合改进的粒子群算法，提出一种特征结构配置重构控制设计方法，解决了故障可能引起的维数不匹配情况下的特征结构配置重构设计，以及基于输出反馈的特征结构配置重构方法不能配置全部闭环特征值而可能引起的系统不稳定性。

(2) 提出一种简单自适应容错控制方法。本文基于指令跟踪发生器理论，进一步研究了简单自适应容错控制方法，分别采用了前向补偿器和反馈补偿器保证了系统的严格正实性；在自适应控制器部分，引入误差积分抑制外部扰动，改进了简单自适应控制器的鲁棒性，并进一步针对 PI 型简单自适应控制律存在着对被控对象要求过严和控制律计算困难等不足，提出一种变步长修正的简单自适应容错控制方法，保证系统输出跟踪理想参考模型的输出。

(3) 提出一种基于神经网络辨识器的容错控制设计方法。这种容错控制方法不要求精确的故障模型和故障参数信息，其基本思想是：利用神经网络建立系统的辨识器模型，作为系统的一个物理冗余通道应用于容错控制中，对故障环节的隔离和重构。本文结合改进的粒子群算法和蚁群信息素思想，提出一种对神经网络的结构和权值并行训练的学习策略，在此基础上，离线建立了神经网络辨识器模型；结合残差序贯概率比假设检验，采用训练好的神经网络辨识器对飞行系统进行在线的故障隔离与自适应重构设计。

(4) 提出一种基于神经网络控制器的自适应容错飞行控制方法。针对非线性

性不确定的飞行控制系统故障，该策略采用反步递推（backsteeping）设计了自适应容错控制律和神经网络权值调整算法，采用 CMAC 同时学习故障后的不确定性和虚拟控制量的导数信息，抵消了故障不确定项对控制的影响，并避免了 Backsteeping 算法的计算膨胀问题。这种容错控制策略为 Control Lyapunov 函数的构造提供了一类较为简便的结构化、系统化方法，保证了系统的输出跟踪和神经网络学习收敛。

7.2 工作展望

本文虽然解决了一些问题和难点，并对提出的算法进行了仿真验证，但由于时间有限，仍有很多值得进一步探讨和深入研究的问题，概括起来主要有以下几个方面：

1. 飞行控制律重构的实时性。飞机的系统状态及参数变化快，这对飞行控制律重构方法的实时性要求较高。本文研究的特征结构配置重构控制虽然可以很好的实现故障系统性能的恢复，但是在实时性上还有待改善；不只特征结构配置重构控制方法存在这一缺陷，很多经典的重构设计方法都存在这一问题。在预知故障较少的情况下，可以通过和多模型自适应、专家系统等控制思想结合的方式加以改善，然而对于突发未知故障或者预知故障较多时，这种方案也将束手无策。如何保证重构方法的快速有效性是始终贯穿容错控制研究的一个难点。

2. 非线性不确定系统的自适应容错控制问题。现代飞行控制系统朝着大型化发展，其结构繁杂，所含子系统较多，并且各个子系统之间互相耦合；同时飞行任务日趋复杂，飞行中易受外部干扰的影响，这些因素都导致故障系统经常表现出非线性以及不确定性的特点，在这种情况下传统的依赖于精确数学模型的容错控制方案将会失效，智能容错控制具有更大的潜力。

神经网络是目前智能控制中的研究热点，研究内容主要集中在快速有效的学习算法、结构优化、保证和评估重构控制律后控制系统的稳定性和收敛性等方面。本文提出的基于蚁群和改进粒子群的神经网络学习算法，虽然实现了网络权值和结构的并行优化，但是这种算法在隐层节点控制因子的更新过程中，并没有使用启发性信息，这是由于目前对神经网络隐层中各神经元的具体物理意义还不清楚造成的，所以，在神经网络的信息处理机制方面，还需要进行更深入的研究；另外，本文采用的反步递推方法，虽然保证了重构后系统的镇定和收敛，也具有较好的实时性，但是这种设计方法对每一个子系统设计一个神经网络，系统维数较高时会使得设计变得庞大和复杂，如何简化控制器的设计成为反步递推设计方案不可回避的问题，值得进一步地深入研究。

另外，在设计形式上，将神经网络等智能方法和传统控制理念结合，吸取两者的优势，是今后容错控制研究的热点问题。

3. 关联系统的容错控制问题。由于时间问题，本文对此并没有进行讨论，但在工程实际中这一问题很有意义。现代系统日趋庞大复杂，系统通常包含多个子系统，并且每个子系统间可能存在着状态、输入、输出间的强耦合。相对于集中控制，分散控制系统更容易实现容错控制。因此复杂大系统的容错控制问题也可以被看作是关联系统间的容错控制问题，对分散大系统/关联系统的容错控制问题的研究有着重要的理论意义和实用价值，是工程实际中重视的问题。

参考文献

- [1] A. Niederlinski, A Heuristic Approach to the Design of Interacting Multivariable Systems, *Automatica*, 1971, 20(7): 691~701
- [2] R. V. Beard, Failure Accommodation in Linear Systems through Self-reorganization, Report MVT-71-1, Man Vehicle Lab, MIT, Cambridge, Massachusetts, 1971
- [3] Siljak D D., Reliable control using multiple control systems. *International Journal of Control*, 1980, 31(2):303~329
- [4] Vidyasagar M., Viswanadham N., Reliable stabilization using a multi-controller configuration. *Automatica*, 1985, 21(5): 599~602
- [5] Vidyasagar M. Some results on simultaneous stabilization with multiple domains of stability. *Automatica*, 1987, 23(6): 535~540
- [6] Gundes A N, Stabilizing controller design for linear systems with sensor or actor failure. *IEEE Transactions on Automatic Control*, 1994, 39(6):1224~1230
- [7] Sebe N and Kitamori T. Reliable stabilization based on multi-compensator configuration. *Proceedings of 12th IFAC World Congress*, 1993, 4, 9~12
- [8] R. Saeks and J. Murrav. Fractional representation, algebraic geometry, and the simultaneous stabilization problem. *IEEE Trans. On Automatic Control*, 1982, 24(4): 895~903
- [9] P. T. Kabamba and C. Yang. Simultaneous controller design for linear time-invariant systems. *IEEE Trans. On Automatic Control*, 1991, 36(1): 106~111
- [10] 张新家, 容错控制理论及其应用, 西北工业大学博士论文, 1990
- [11] Le Gorrec Y, Magni J and Chiappa. Modal multimodel control design approach applied to aircraft autopilot design. *Journal of Guidance, Control and Dynamics*, 1998, 21(1):77~83
- [12] 孙金生, 李军, 徐蕾, 王执铨, 动态大系统的分散容错控制, *南京理工大学学报*, 1998, 15(2): 117~120
- [13] Joshi S M. Failure-accommodation control of large flexible spacecraft. *Proceedings of American Control Conference*, Seattle, WA, 1986, 156~161
- [14] 刘瑞民, 俞金寿, 林家骏, 基于 Riccati 方程的双线性系统容错设计-执行

- 器故障情况[A]. In: 张嗣瀛等, 2000 年中国控制与决策学术年会论集[C]. 沈阳: 东北大学出版社, 2000, 427~431
- [15] 葛建华, 孙优贤, 周春辉. 故障系统容错能力判别的研究信息与控制, 1989, 18(4): 8~11
- [16] Ye Y Z. Fault tolerant pole assignment for multivariable systems using a fixed state feedback, 控制理论与应用, 1993, 10(2): 212~218
- [17] 唐小静, 容错控制理论及其应用研究, 西北工业大学博士论文, 2000
- [18] R. V. Beard. Failure accommodation in linear systems through self-reorganization. Report MVT-71-1, Man Vehicle Lab, MIT, Cambridge, Massachusetts, 1971
- [19] Zhang Xiaodong. Fault diagnosis and fault-tolerant control for nonlinear systems. Ph.D, 2002
- [20] 王德军, 故障诊断与容错控制方法研究, 吉林大学博士论文, 2004
- [21] Frank P.M. fault diagnosis in dynamic using analytical and knowledge based redundancy-a survey and some new results[J], Automatica, 1990, 26(3): 459~474.
- [22] 周东华, 叶银忠, 现代故障诊断与容错控制[M], 北京: 清华大学出版社, 2000
- [23] 周东华, 王杜增, 故障诊断技术综述[J], 化工自动化及仪表, 1998, 25(1): 58~62
- [24] 罗小元, 几类非线性系统鲁棒故障诊断及容错控制研究, 燕山大学博士论文, 2005
- [25] 张翠侠, 基于观测器的非线性系统故障诊断研究, 贵州大学硕士论文, 2008
- [26] Patton R, Frank P, M. Clark, R. N, Fault Diagnosis in Dynamic Systems[M], Theory and Application, Englewood Cliffs, NJ: Prentice Hall, 1989
- [27] S. Dash, V. Venkatasubramanian, Challenges in the industrial application of fault diagnosis systems[J], Computers and chemical engineering, 2000, 24(8): 785~791
- [28] Gao Z, Antsaklis P J. Stability of the pseudo-inverse method for reconfigurable control systems. International Journal of Control, 1991, 53: 502~529
- [29] Jiang J. Design of reconfigurable control systems using eigenstructure assignments. International Journal of Control, 1994, 59(2): 395~410
- [30] Sobel K., Kaufman H., Mabius L., Implicit adaptive control for a class of MIMO systems[J], IEEE Trans Aerospace and Electronic Systems, 1982,

- 18(5):576~590
- [31] Barkana I., Kaufman H., Balas M., Model reference adaptive control of large structural systems, *AIAA Journal of Guidance*, 1983, 6(2):112~118
- [32] Barkana L., Kaufman H., Some applications of direct adaptive control to large structural Systems, *AIAA Journal of Guidance*, 1984, 7(6):717~724
- [33] Itzhak Barkana, On gain conditions and convergence of simple adaptive control, *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Austin, Texas, 11~14 August 2003, pp:5327.1~14
- [34] Itzhak Barkana, Gain conditions and convergence of simple adaptive control, *International journal of adaptive control and signal processing*, Published online in wiley InterScience (www.interscience.wiley.com), 2005, pp:1~28
- [35] Itzhak Barkana, Simple adaptive control for non-minimum phase autopilot design, *Journal of guidance, control, and dynamics*, 2005, 28(4):631~639
- [36] Kaufman H., Barkana I, Sobel K.M. *Direct Adaptive Control Algorithms: Theory and Applications*, Second Edition. Berlin: Springer-Verlag, 1998
- [37] Abderrazak Idrissi Belkharraz, *Fault Tolerant Control for Aircraft Control Surface Failures*, a Dissertation for the Doctor's Degree, The City University of New York, 2003
- [38] A. E. Ashari, A. K. Sedigh, M.J. Yazdanpanah, Output feedback reconfigurable controller design using eigenstructure assignment:post-fault order change[C], *Proceedings of International Conference on Control and Automation*, Budapest, 2005, pp. 474~479
- [39] Naerndra K, Parthasarathy. *Identification and Contorl of Dynamical Systems Using Neural Networks*. *IEEE Transactions on Neural Networks*, 1990, 1(1):4~7
- [40] K.S.Narendra. *Neural Networks for Control: Theory and Practice*. *Proceedings of the IEEE*, 1996, 84(10):1385~1406
- [41] 吴海桥, 现代大型客机故障诊断专家系统的研究与开发, 南京航空航天大学博士论文, 2002
- [42] 贾明兴, 不确定系统鲁棒故障诊断, 东北大学博士论文, 2003
- [43] 钱华明, 故障诊断与容错技术及其在组合导航系统中的应用研究, 哈尔滨工程大学博士论文, 2004
- [44] BAR-KANA I, *Adaptive control: a simplified approach*[J]. *Control and Dynamic Systems*, Academic Press, 1987, 25:187~235

- [45] BAR-KANA I, Parallel feedforward and simplified adaptive control[J].Int J Adaptive control and signal Processing, 1987, 1(2):95~109
- [46] BAR-KANA I, Adaptive control: a simplified approach[J], Control and Dynamic Systems, Academic Press, 1987, 25:187~235
- [47] Kanellakopoulos I, Kokotovic P, Morse A S. Systematic Design of Adaptive Controllers for Feedback Linearizable Systems. IEEE Trans on AC, 1991, 36(11): 1241~1253
- [48] Freeman R A, Kokotovic P V. Backstepping Design of Robust Controllers for a Class of Nonlinear Systems. Proc of the IFAC Nonlinear Control Systems Design Symposium, Bordeaux: 1992, 307~312
- [49] Ola Harkegard, S. Torkel Glad, A Backstepping. Design for Flight Path Angle Control. Proceedings of the 39th IEEE Conference on Decision and Control, Dec 2000, 3570~3575
- [50] Taeyoung Lee, Youdan Kim. Nonlinear Adaptive Flight Control Using Backstepping and Neural networks Controller. AIAA Journal of Guidance, Control, and Dynamics, 2001, 24(4): 675~682
- [51] Marc L, Steinberg, Anthony B. Page. Nonlinear Adaptive Flight Control with A Backstepping Design Approach. AIAA Guidance, Navigation, and Control Conference and Exhibit, 1998, Aug: 728~738
- [52] Marc L, Steinberg. Comparison of Intelligent, Adaptive, and Nonlinear Flight Control Laws. Journal of Guidance, Control, and Dynamics, 2001, 24(4): 693~699
- [53] M. Sharma, D. Ward, Barron Associates Inc., Charlottesville, VA; J. Farrell. Backstepping Flight Control Using On-Line Function Approximation. AIAA Guidance, Navigation, and Control Conference and Exhibit, 2003, Aug
- [54] Zhang Youmin, Jiang Jin, An interacting multiple-model based fault detection, diagnosis and fault-tolerant control approach[C] //Proceedings of the IEEE Conference on Decision and Control Piscataway, NJ IEEE,1999, 4: 3593~3598
- [55] Andry A N, Shapiro Jr E Y, Chung J C, Eigenstructure Assignment for Linear Systems[J], IEEE Trans. on Aerospace and Electronic ,1983, 19(3): 711~729
- [56] Jiang J, Design of reconfigurable control systems using eigenstructure assignments[J], Int J contr, 1994, 59 :395~410
- [57] Sobel KM, Shapiro EY: “Application of eigenstructure assignment to flight control design: Some extensions”, Journal of Guidance, Control and Dynamics,

- 1987, 10(1): 73~81
- [58] Sobel KM, Yu W, Lallman FJ: "Eigenstructure assignment with gain suppression using eigenvalue and eigenvector derivatives", *Journal of Guidance, Control and Dynamics*, 1990, vol.13, no.6
- [59] K.J. Hunt, D.Sbarbaro, R. Zbikowski, P.J. Gawthrop, *Neural Networks for Control System – A survey Automatica*, Vol.28, No.6, pp:1083~1112
- [60] Baumgarten G, Buchholz J and Heine W, A new reconfiguration concept for flight control systems. AIAA-95-3176-CP: 9~19
- [61] Marcello R, Napolitano, Jose L et al. on-line Learning Neural and Fuzzy Logic Controllers for actuator failure accommodation in flight control systems: AIAA-GNCC-1997:390~400
- [62] Barton et al., R. Applications of Polynomial Neural Networks to FDIE and Reconfigurable Flight Control, *Proceedings of the IEEE National Aerospace and Electronics Conference*, 1990, p506~519
- [63] W.L.Baker, J. A. Farrell, Learning Augmented Flight Control for High Performance Aircraft. *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, 1991, p347~358
- [64] Dong-Ho Shin and Youdan Kim, Reconfigurable Flight control System design using adaptive neural networks, *IEEE Transaction on control systems technology*, 2004, 12(1):87~102
- [65] Campa G, Fravolini M.L., and Napolitano M.R., A stochastically optimal feedforward and feedback technique for flight control systems of high performance aircrafts, *American Control Conference*, 2004
- [66] 喻宗泉, 喻晗, 神经网络控制[M], 西安电子科技大学出版社, 2009
- [67] 朱大齐, 史慧, 人工神经网络原理及应用[M], 科学出版社, 2006
- [68] J.S.Albus, A New Approach to Manipulator Control: the Cerebellar Model Articulation Controller(CMAC), *Trans, ASME, J.Dyn.Syst.Meas. Contro.*, 1975, 97(3):220~227
- [69] Taeyoung Lee, Youdan Kim. Nonlinear Adaptive Flight Control Using Backstepping and Neural networks Controller. *AIAA Journal of Guidance, Control, and Dynamics*, 2001, 24(4): 675~682
- [70] 刘树棠, 韩崇昭, 自适应逆控制[M], 西安交通大学出版社, 1999
- [71] Kanellakopoulos I, Kokotovic P, Morse A S. Systematic Design of Adaptive Controllers for Feedback Linearizable Systems. *IEEE Trans on AC*, 1991,

- 36(11): 1241~1253
- [72] Freeman R A, Kokotovic P V. Backstepping Design of Robust Controllers for a Class of Nonlinear Systems. Proc of the IFAC Nonlinear Control Systems Design Symposium, Bordeaux: 1992, 307~312
- [73] Ola Harkegard, S. Torkel Glad, A Backstepping. Design for Flight Path Angle Control. Proceedings of the 39th IEEE Conference on Decision and Control, Dec 2000, 3570~3575
- [74] Taeyoung Lee, Youdan Kim. Nonlinear Adaptive Flight Control Using Backstepping and Neural networks Controller. AIAA Journal of Guidance, Control, and Dynamics, 2001, 24(4): 675~682
- [75] Marc L, Steinberg, Anthony B. Page. Nonlinear Adaptive Flight Control with A Backstepping Design Approach. AIAA Guidance, Navigation, and Control Conference and Exhibit, 1998, Aug: 728~738
- [76] Marc L, Steinberg. Comparison of Intelligent, Adaptive, and Nonlinear Flight Control Laws. Journal of Guidance, Control, and Dynamics, 2001, 24(4): 693~699
- [77] M. Sharma, D. Ward, Barron Associates Inc., Charlottesville, VA; J. Farrell. Backstepping Flight Control Using On-Line Function Approximation. AIAA Guidance, Navigation, and Control Conference and Exhibit, 2003, Aug
- [78] Saverio M., Backstepping design for controlling Lorenz chaos. In Proc of the 36th IEEE CDC San Diego, CA, pp.1500~1501
- [79] Harkegard O., and Glad S.T., Flight control deign using backstepping. Technical Report LiTH-ISY-R-2323, Department of Electrical Engineering, Linkopings university, 5E-58183, Linkoping, Sweden, Dec.2000
- [80] 王福利, 张颖伟, 容错控制[M], 东北大学出版社, 2003
- [81] 张伟, 电液力伺服系统控制方法研究, 西北工业大学博士论文, 2009
- [82] 何军红, 复杂系统可靠性和容错控制及其应用, 西北工业大学博士论文, 2001
- [83] 安世奇, 孙一康, 王京, 简单自适应控制的算法与发展[J], 电机与控制学报, 2004, 8(3): 263~267
- [84] Capriglioner D, Iguori C, and Pietrosanto A, Analytical Redundancy for Sensor Fault Isolation and Accommodation in Public Transportation and measurement, IEEE Transaction on, 2004, 53(4):1457~1462
- [85] Polycarpou M, Xiaodong Zhang, Xu R, Yanli Yang, Chiman Kwan, A Neural

-
- Network Based Approach to Adaptive Fault Tolerant Flight Control [C]//Intelligent Control, Proceedings of the IEEE International Symposium on, 2004:61~66
- [86] Napolitano Marcello R and Giovanni Silvertri, Sensor Validation using Hardware-Based on-line Learning Neural Networks[J], IEEE transactions on aerospace and electronic systems, 1998, 34(2):56~66
- [87] Mattern Duance L and Jaw Link C, using Neural Networks for Sensor Valiation[C]//AIAA Joint propulsion conference &exhibit, Cleveland OH,1998:1~12
- [88] Marinaro M, Scarpetta, S, On-learning in RBF Neural Networks: a Stochastic Approach[J], Neural Networks, Elsevier Science, 2000, 13(7):720~729
- [89] Wang Lipo, Liu Bing, and Wan Chunru, A novel RBF Neural Network with Fast Training and Accurate Generalization[C]//Computational and Information Science First International Symposium, Spring-Verlag Berlin Heidelberg 2004, Vol.33, pp:166~171
- [90] Anastasios Tsoutis, An analusis of the Joint Strike Fighter Automatic Logistics System[D],USA: University of Florida, 2000
- [91] Scheuren W. J, Caldwell K. A., Goodman G. A,et al, Joint Strike Fighter Prognostics and Health Manager[R], 34th Joint Propulsion Conference & Exhibit[C], Cleveland, 1998
- [92] McCulloch W, Pitts W., A Logical Calculus of the Ideas Immanent in Nervous Activity. Bulletin of Mathematical Biophysics, 1943, Vol.5, pp.115~133
- [93] D.O.Hebb. The Organization of Behavior[M]. Wiley, New York, 1949
- [94] F.Rosenblatt. The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain[J]. Psychology Review, 1958, Vol.65, No.6, pp.386~408
- [95] B.Widrow, M.E.Hoff., Adaptive Switching Circuits[C], IRE WESCON Convention Record, New York, 1960, Vol.4, pp.96~104
- [96] Marvin Minsky, Seymour Papert. Perceptrons[M], Cambridge: MIT Press, 1969
- [97] T.Kohonen. Correlation Matrix Memories[J], IEEE Transactions on Computers, 1972, Vol.21, pp.352~359
- [98] S.A.Grossberg. Studies of Mind and Brain[M], Holland: Reidel Press, 1982
- [99] S.A.Grossberg, Contour Enhancement Short Term. Memory and constancies in Reverberating Neural Networks[J], Studies in Applied Mathematics, 1973, Vol.52, No.3, pp.213~257

-
- [100] .Fukushima. Neocognition: A Self-organizing Neural Network Model for A Mechanism of Pattern Recognition Unaffected by Shift in Positon[J], Biol Cybernetics, 1980, Vol.36, pp.193~202
- [101] 王青, 孙蕊, 基于特征结构配置的导弹可重构控制器设计, 北京航空航天大学学报, 2008, 34(3) : 304~306
- [102] 李中健, 大包线飞行控制系统鲁棒设计研究, 西北工业大学博士论文, 2000
- [103] 张明涛, 安锦文, 李中健, 重构飞行控制技术研究综述[J], 飞行力学, 2009, 27(2): 7~10
- [104] 王群伟, 无人机飞控系统控制律重构技术研究, 西北工业大学硕士论文, 2007
- [105] Johnson E.M, Dowla F U, Goodman D.M. Backpropagating Learning for Multilayer Feed-forward Neural Networks using the Conjugate Gradient Method[J]. Int J Neural System, 1991, 2(4): 291~302
- [106] Montana D J, Davis L. Training Feed-forward Neural Networks using Genetic Algorithms[C]. In IJCAI, Proceedings of the International Joint Conference on Artificial Intelligence, 1989, Vol.1, pp.762~767
- [107] Jarmo Ilonen, Kamarainen Jon, Lampinen J. Differential Evolution Training Algorithm for Feed-forward Neural Networks[J], Neural Processing Letters, 2003, Vol.17, pp.93~105
- [108] 陈新海, 李言俊, 周军, 自适应控制及应用,西北工业大学出版社, 2003
- [109] 刘小雄, 飞行控制系统故障隔离与自适应重构技术研究, 西北工业大学博士论文, 2006
- [110] Marcello R. Napolitano, Steve Naylor, Charles Neppach, On-line Learning Non-Linear direct Neuro Controllers for Restructurable Control Systems[J], AIAA-94-3643-CP
- [111] Giampiero Campa, Mario L. Fravolini, Marcello R. Napolitano, Neural Networks-Based Sensor Validation for the Flight Control System of a B777 Research Model , Proceedings of the American Control Conference Anchorage, 2002
- [112] Marcello R. Napolitano, Younghwan An, Brad A. Seanor, A fault tolerant flight control system for sensor and actuator failures using neural networks, Aircraft Design 3 (2000) 103~128, 2006
- [113] J.S.Pei, E.C.Mai. A Heuristic Prototype-based Initialization Approach for Approximating Engineering Nonlinear Functions using Multilayer Feedforward Neural Networks[M], Neurocomputing, 2006
- [114] Vogl T P, Mangis J K, Rigler J K, *et al.* Accelerating the Convergence of the

-
- Back Propagation Method[J], Biological Cybernetics, 1988, 59(3): 257~263
- [115] Chan L W, Fallside F. An Adaptive Training Algorithm for Back Propagation Networks[J], Computers, Speech and Language, 1987, No.2, pp.205~218
- [116] Jacobs R A. Increased Rates of Convergence Through Learning Rate Adaptation[J], Neural Networks, 1988, 1(4): 295~307
- [117] Hsin H C, Li C C, Sun M, *et al.* An Adaptive Training Algorithm for Back-propagation Neural Network[J], IEEE Transactions on System, Man and Cybernetics, 1995, 25(4): 512~514
- [118] Magoulas G D, Vrahatis M N, Androulakis G S. Effective Back-propagation with Variable Step Size[J]. Neural Networks, 1997, 10(1): 69~82
- [119] Kaminura R, Nokanishi S. Hidden Information Maximization for Feature Detection and Rule Discovery[J], Network Computation in Neural System, 1995, 6(6): 577~602
- [120] Friedrich B K, Bamrann F. A Learning Algorithm for Multilayered Neural Networks Based on Linear Least Square Problems[J]. Neural Networks, 1993, 4(6): 127~602
- [121] Hagan M T, Menhaj M B. Training Feedforward Networks with the Marquardt Algorithm[J], IEEE Transactions on Neural Networks, 1994, 5(6): 989~993
- [122] Johnson E.M, Dowla F U, Goodman D.M. Backpropagating Learning for Multilayer Feed-forward Neural Networks using the Conjugate Gradient Method[J], Int J Neural System, 1991, 2(4): 291~302
- [123] Moller M F. A Scale Conjugate Gradient Algorithm for Fast Supervised Learning[J], Neural Networks, 1993, 4(6): 525~533
- [124] Mzard M, Nadal J P. Learning in Feed-forward Layered Networks: the Tiling Algorithm[J], J.Phys. A: Maths.Gen, 1989, 22(8): 2191~2203
- [125] Frean M. The Upstart Algorithm: A Method for Constructing and Training Feed-forward Neural Networks[J], Neural computation, 1990, 2(2): 198~209
- [126] Marcello R. Napolitano, Steve Naylor and Ching I Chen, AIRCRAFT FAILURE DETECTION AND IDENTIFICATION USING NEURAL NETWORKS, AIAA-93-3869-CP, 1517~1526
- [127] 刘小雄, 章卫国, 李广文, 基于智能解析冗余的容错飞控系统的设计, 传感技术学报, 2007, 20(8): 1912~1915
- [128] 宁东方, 神经网络在飞控系统健康管理技术中的应用研究, 西北工业大学博士论文, 2008
- [129] Seungyong Hyung and Youdan Kim, Reconfigurable flight control system design using discrete model reference adaptive control, AIAA Guidance, Navigation, and Control Conference and Exhibit, San Francisco, California,

- AIAA 2005-634315-18 2005, pp:1~9
- [130] Felipe M Pait, On the design of direct adaptive controllers, Proceedings of the 40th IEEE conference on decision and control, USA, 2001, pp: 734~74
- [131] R. T. Rysdyk and A. J. Cause, Adaptive model inversion flight control for tilt-rotor aircraft, J. Guidance, Control, Dynamitic, 1999, 22(3): 402~407
- [132] 周姝春, CMAC 学习性能及泛化性能研究综述, 计算机仿真, 2005, 22 (6): 5 ~7
- [133] 罗忠, 谢永斌, 朱重光, CMAC 学习过程收敛性的研究, 自动化学报, 1997, 23(4): 455~461
- [134] Kokotovic P.V., The Joy of Feedback: Nonlinear and Adaptive, IEEE Control Systems, 1992,12(3):7~17
- [135] B. S. Kim and A. J. Calise, Nonlinear adaptive flight control using neural networks, J. Guidance, Control, Dynamitic, 1997, 20(1): 26~33
- [136] A.J.Calise, S.Lee, and M.Sharma, Direct adaptive reconfigurable control of tailless fighter aircraft, AIAA-98-4107, 1998, pp:88~98
- [137] L. Lightbody, G.W. Iwin, Direct neural model reference adaptive control, Proc. Pt. D. Control Theory and Application, 142(1):31~43, 1995
- [138] Doze Ritonja, Drago Dolinar, and Bojan Grcar, Simple adaptive control for stability improvement, proceedings of the IEEE international Conference on control applications, 2001, pp: 29~36
- [139] K. Patric, Simson, Artificial neural network systems: Foundations, paradigms, applications and implementations, Pergamon Press, 1990
- [140] Robert J. Fuentes, Direct adaptive disturbance accommodation, proceedings of the 39th IEEE Conference on Decision and Control Sydney, Australia December, 2000, pp:4921~4926
- [141] Zhuo Wang, Ping Li, Shuo Guo, Adaptive inverse control for nonlinear systems based on RBF neural network, Intelligent Control and Automation, WCICA Fifth World Congress on, 2004, Vol.1 :485~487
- [142] Johansson R., and Robertsson A, Observer-based strict positive real (SPR) feedback control system design, Automatica, 2002, Vol. 38:1557~1564
- [143] Yagiz Nurkan, Hacioglu Yuksel, Backstepping control of a vehicle with active suspensions, Control Engineering Practice, v 16, n 12, December, 2008, p 1457~1467
- [144] Ozbay U., Zergeroglu E., Sivrioglu S., Adaptive backstepping control of

- variable speed wind turbines, *International Journal of Control*, v 81, n 6, June, 2008, p 910~919
- [145] H. Noura, D. Sauter, F. Hamelin, D. Theilliol. Fault-tolerant Control in Dynamic System: Application to A Winding Machine. *IEEE Control System Magazine*, 2000: 33~49
- [146] Y. M. Zhang, Jin Jiang, Design of Proportional-integral Reconfigurable Control Systems via Eigenstructure Assignment. *Proceedings of the American Control Conference*. 2000, 6: 3732~3736
- [147] Y. Zhang, J. Jiang. Fault Tolerant Control Systems Design with Consideration of Performance Degradation. *Proceedings of the American Control Conference*, Arlington, 2001, 2693~2699
- [148] C Y Huang, R F Stengel. Reconstructurable control using proportional-integral implicit model following[J], *Journal of Guidance, Control and Dynamic*, Vol. 13, No. 2, 1990:303~309
- [149] Zhang Xiaodong, Marios M. Polycarpou, Thomas Parisini. A robust detection and isolation scheme for abrupt and incipient faults in nonlinear systems. *IEEE Trans. on Autom. Contr*, 2002, 47(2):576~593
- [150] D. H. Zhou, P.M. Frank. Fault diagnosis and fault tolerant control. *IEEE Trans. On Aerospace and Electronic Systems*, 1998, 34(2): 420~427
- [151] Ching-Tsan Chiang, Chun-shin Lin, CMAC with general basis functions. *Neural Networks*. 1996, 9(7):1199~1211
- [152] Francisco J. Gonzalez-Serrano, Anibal R. Figueiras-Vidal, Antonio Artes Rodriguez Fourier analysis of the generalized CMAC neural network. *Neural Networks*. 1998, 11(3):391~396
- [153] Aleksander Kolcz, Nigel M. Allinson. Basis function models of the CMAC network. *Neural Networks*. 1999, 12(4):1071~126
- [154] A. E. Ashari, A. K. Sedigh, M.J. Yazdanpanah, Output feedback reconfigurable controller design using eigenstructure assignment: post-fault order change[C], *Proceedings of International Conference on Control and Automation*, Budapest, 2005, pp. 474~479
- [155] Srinathkumar, S., Eigenvalue /Eigenvector Assignment Using Output Feedback[J], *IEEE Transactions on Automatic Control*, 1978, Vol. 23, pp. 79~81

- [156] J. Jiang, Design of reconfigurable control systems using eigenstructure assignments[J], Int J. Control. 1994, vol.59, pp. 395~410
- [157] A. E. Ashari, A. K. Sedigh, M.J. Yazdanpanah, Reconfigurable control system design using eigenstructure assignment: static, dynamic and robust approaches, Int J. control. 2005, 78(13): 1005~1016
- [158] Hee-Seob Kim, Youdan Kim, Partial eigenstructure assignment algorithm in flight control system design[J], IEEE Transactions on Aerospace and Electronic systems, 1999, 35 (4): 1043~1048
- [159] Kenneth M. Sobel, E.Y. Shapiro, Application of eigenstructure assignments to flight control design: some extensions[J], Journal of Guidance, Control and Dynamic, 1987, 10(1): 73~81
- [160] L.F. Faleiro, R.W. Pratt, Multi-objective eigenstructure assignments with dynamics flight control augmentation systems[J], Journal of Guidance, Control and Dynamic, 1996, 19(1): 1~10
- [161] Ren Zhang, Tang Xiaojing, Chen Jie, Reconfigurable control system design by output feedback eigenstructure assignments[J], Control Theory and Application, 2002, vol 19(3): 356~362
- [162] I.K. Konstantopoulos, P.J. Antsaklis, Optimization approach to robust eigenstructure assignment[J], IEEE Pro-Control Theory Appl, 1999, 146(6): 561~565
- [163] 刘小雄等, 直接自适应鲁棒飞行控制技术研究, 西北工业大学学报, 2008, 26(3): 341~345
- [164] A. Colomi, M. Dorigo, V. Maniezzo, Distributed optimization by ant colonies[C], The 1st European Conference on Artificial Life, 1991, pp. 134~142, 1991
- [165] Kumpati S. Narendra and Jeyendran Balakrishnan, Adaptive control using multiple models[J], IEEE transactions on automatic control, 1997, 42(2): 171~187
- [166] McCulloch W, Pitts W. A Logical Calculus of the Ideas Immanent in Nervous Activity. Bulletin of Mathematical Biophysics, 1943, Vol.5, pp. 115~133
- [167] D.O. Hebb, The Organization of Behaviour[M], Wiley, New York, 1949
- [168] Hopfield J J. Neurons with Graded Response Have Collective Computational Properties Like those of Two-state Neurons[C]. Proceedings of the National Academy of Science, USA (Biophysics), 1984, Vol.81, pp. 3088~3092
- [169] Mendes R, Cortez P, Rocha M, *et al.* Particle Swarms for Feedforward Neural

- Network Training[C]. Proceedings of the 2002 International Joint Conference on Neural Networks, 2002, pp.1985~1899
- [170] 骆再飞, 滑模变结构理论及其在交流伺服系统中的应用研究, 浙江大学博士学位论文, 2003.7
- [171] R. Isermann. Process Fault-detection Based on Modeling and Estimation Methods: A Survey. *Automatica*, 1984, 20: 387~404
- [172] D. H. Zhou, P.M. Frank, Fault diagnosis and fault tolerant control, *IEEE Trans, On Aerospace and Electronic Systems*, 1998, 34(2): 420~427
- [173] R. J. Patton, J. Chen, A review of parity space approaches to fault diagnosis, *Proceedings of IFAC Fault Detection, Supervision and Safety for Technical Processes*, Baden, Germany, 1991:65~81
- [174] 闻新, 张洪钺, 周露, 控制系统的故障诊断和容错控制, 机械工业出版社, 2000
- [175] H. Scheider, Implementation of a Fuzzy concept for supervision and fault detection of robots EUFIT, *First European Congress on Fuzzy and Intelligent Technologies*, Aachen, Germany, 2002:775~780
- [176] Shapiro. E. Y., Chung. J.C, Application of eigenvalue/eigenvector assignment by constant output feedback to flight systems design, In *Proc.15th Annu. Conf. Information Sciences and Systems*, 1981, pp:164~169
- [177] Shapiro. E. Y., Chung. J.C., Constrained eigenvalue/eigenvector assignment-Application to flight control systems, In *Proc.19th Annu. Conf. Communication Control and Computing*, 1981, pp:320~328
- [178] K.Konstantopoulos, P.J. Antsaklis, Eigenstructure assignment in reconfigurable control system , *Technical Report, Interdisciplinary Studies of Intelligent Systems*, University of Notre Dame, 1996
- [179] Tong S, Li Y, Direct adaptive fuzzy backstepping control for a class of nonlinear systems, *International Journal of Innovative Computing&Information Control*, 2007, 3(4):887~896
- [180] Jiang B, Staroswiecki M, Cocquempot V, Fault accommodation for nonlinear dynamic systems, *IEEE Trans, on Automatic Control*, 2006, 51(9): 1578~1583
- [181] Jin X Z, Yang G H, Robust adaptive fault-tolerant compensation control with actuator failures and bounded disturbance, *Acta Automatica Sinica*, 2009, 35(3): 305~309
- [182] Zhang K, Jiang B, Shi P, Fast fault estimation and accommodation for

- dynamical systems, IET Control Theory & Applications, 2009, 3(2): 189~199
- [183] Tang X D, Tao G, Joshi S M, Adaptive actuator failure compensation for nonlinear MIMO systems with an aircraft control application, Automatica, 2007, 43(11): 1869~1883
- [184] Ye D, Yang G H, Adaptive fault-tolerant tracking control against actuator faults with application to flight control, IEEE Trans, on Control Systems Technology, 2006, 14(16):1088~1096
- [185] Meng Lingya, Jiang Bin, Backstepping-based active fault-tolerant control for a class of uncertain SISO nonlinear systems, Journal of Systems Engineering and Electronics, 2009, 20(6):1263~1270

博士在读期间发表论文和参加科研情况说明

发表论文情况:

- [1] An adaptive backstepping design for longitudinal flight path control, Proceedings of the 7th World Congress on Intelligent Control and Automation (WCICA 2008), ISBN: 978-1-4244-2113-8, 2008.6, EI 收录: 083911600953, ISTP 收录: 000259965704068 (第一作者)
- [2] 基于特征结构配置和直接自适应的侧向飞行控制律设计, 测控技术, 2009.8, Vol.28, No.8 (第一作者)
- [3] 可重构飞行控制律设计的混合特征结构配置方法, 测控技术, 2009.11, Vol.28, No.11 (第一作者)
- [4] 模糊控制与特征结构配置方法在飞行控制律设计中的应用, 弹箭与制导学报, 2006, Vol.26, No.2 (第一作者)
- [5] 基于自适应逆的鲁棒容错飞行控制方法研究, 计算机仿真, 已录用 (第一作者)
- [6] 非匹配不确定非线性系统的鲁棒输出跟踪, 西北工业大学学报, 2006.10, Vol.24, No.5, EI 收录: 070310361906 (第三作者)
- [7] 基于特征结构配置的非线性控制器在直接力控制中的应用, 弹箭与制导学报, 2006, Vol.26, No.3 (第三作者)
- [8] 直升机旋翼协调加载自适应最优解耦控制, 中国机械工程, 2007. 11, Vol.18, No.22, EI 收录: 075210995873 (第三作者)
- [9] 电液负载模拟器的自适应最优控制, 机床与液压, 2007.6, Vol.35, No.6 (第三作者)
- [10] 基于自适应反演的电液伺服系统设计, 机床与液压, 2007.5, Vol.35, No.5 (第三作者)

参加科研情况:

××地面故障诊断专家系统	中航集团某厂	2006~2010
××飞行控制故障诊断与控制律研究	中航集团某研究所	2007~2009
××控制律优化设计技术研究	中航集团某研究所	2008~2009

致 谢

本论文是在尊敬的导师章卫国教授的悉心指导下完成的。值此论文完稿之际，谨向章老师致以崇高的敬意和衷心的感谢！

特别感谢章老师在我博士学习期间，对我的辛勤培养、殷切教诲和无私的关心帮助。从博士论文的选题到论文的各个研究阶段，章老师都给予了悉心的指导，在此表示诚挚的感谢。

感谢我的父母多年来的养育之恩，他们永远是我力量的源泉，鼓舞我前行、激励我前进。在我的成长过程中，父母为我所费心血难以计量，如今女儿即将完成学业，谨以此作为今年的父亲节礼物，祝亲爱的爸妈身体健康。感谢我的两个姐姐，在我漫长的求学生涯中，她们总是默默无闻的关心我、支持我，始终是我坚强的后盾。对家人的恩情，我尽所有努力也无以回报其万分之一，在此给远在家乡的亲人们献上我深深的祝福。

感谢刘小雄、李广文副教授，在我的论文研究和科研工作中，他们给了我很多帮助和建议，传授了许多学习经验，使我在学习过程中能够克服诸多困难。感谢师兄宁东方同学和同级博士生张伟、王振华、史静平同学，在博士学习中，我们互相帮助、互相促进、互相鼓励。感谢师弟师妹——章萌、孙勇、朱江乐、张竞凯、池程芝、聂瑞、李丹、赵红梅、李丽莉等，在共同学习中，我们互相关心、情同手足，组成了一个欢乐的大家庭。在我的学习和生活中，给予我帮助的人太多太多，感谢所有给过我帮助和支持的亲人、老师、同学和朋友们，祝他们幸福快乐。

西北工业大学

学位论文知识产权声明书

本人完全了解学校有关保护知识产权的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属于西北工业大学。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。本人允许论文被查阅和借阅。学校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。同时本人保证，毕业后结合学位论文研究课题再撰写的文章一律注明作者单位为西北工业大学。

保密论文待解密后适用本声明。

学位论文作者签名：_____

指导教师签名：_____

年 月 日

年 月 日

西北工业大学

学位论文原创性声明

秉承学校严谨的学风和优良的科学道德，本人郑重声明：所呈交的学位论文，是本人在导师的指导下进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容和致谢的地方外，本论文不包含任何其他个人或集体已经公开发表或撰写过的研究成果，不包含本人或其他已申请学位或其他用途使用过的成果。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表明。

本人学位论文与资料若有不实，愿意承担一切相关的法律责任。

学位论文作者签名：_____

年 月 日