容错与冗余技术

容错控制的研究虽然面临着空前的挑战，但近些年来，相关研究领域，如鲁棒控制理论，　模糊控制，神经网络控制研究的不断深入和发展，也给容错控制的研究带来了良好的机遇，　提供了充分的条件。

而计算机控制技术、人工智能等技术的飞速发展，使得容错控制技术在实际工程中应用的可能性变得越来越大。

1.1 容错概念的提出

提高系统的可靠性一般有两种办法：　1、采用缜密的设计和质量控制方法来 尽量减少故障出现的概率　。2、以冗余资源为代价来换取可靠性。

利用前一种方法来提高系统的可靠性是有限的，要想进一步的提高必须采用容错技术。

容错控制技术在国外发展的比较早，是由冯·诺依曼提出的。随着八十年代微型计算机的迅速发展和广泛应用，　容错技术也得到了飞速的发展，容错技术被应用到各个环境中。

我国的容错技术现在发展的也很迅速，一些重要的工作场合如航天、电厂等现在都采用了容错技术。

所谓 容错 ：就是容许错误，是指设备的一个或多个关键部分法生故障时，能够自动地进行检测与诊断，并采取相应措施，保证设备维持其规定功能，或牺牲性能来保证设备在可接受范围内继续工作。

错误一般分为两类　：第一类是先天性的固有错，如元器件生产过程中造成的错、 线路与程序在设计过程中产生的错。　这一类的错误

需对其拆除、更换或修正，是不能容忍的。第二类的错后天性的错，它是由于设备在运行中产生了缺陷所导致的故障。这种故障有瞬时性、间歇性和永久性的区别。

容错技术是提高系统可靠性的重要途径。常采用的容错方法有硬件容错、软件容错、信息容错和时间容错。

1.1.1 智能容错的定义

智能容错 IFT(Intelligent Fault-Tolerance)：就是设备在运行过程中一个或多个关键部件发生故障或即将发生故障之前，利用人工智能理论和方法，通过采取有效措施，对故障自动进行补偿、抑制、消除、修复，以保证设备继续安全、高效、可靠运行，或以牺牲性能损失为代价，保证设备在规定的时间内完成其预定功能。

智能容错技术的构成方法可以采用以下三步来实现：

（1）建立系统的设计目标；

（2）设计智能容错处理机构；

（3）根据设计目标对所作的设计进行评价，如果满足目标则设计成功，否则将返回第二步进行重新设计，直到满足设计目标要求。

硬件智能容错 HIFT (Hardware Intelligent Fault Tolerant) 主要采用硬件冗余技术。其基本思想是对设备的关键部件配备多重相似或相同部件，一旦检测和诊断出设备发生故障就可以立刻切换到备份部件，以达到故障容错的目的。图 1 所示为二冗余结构原理图：

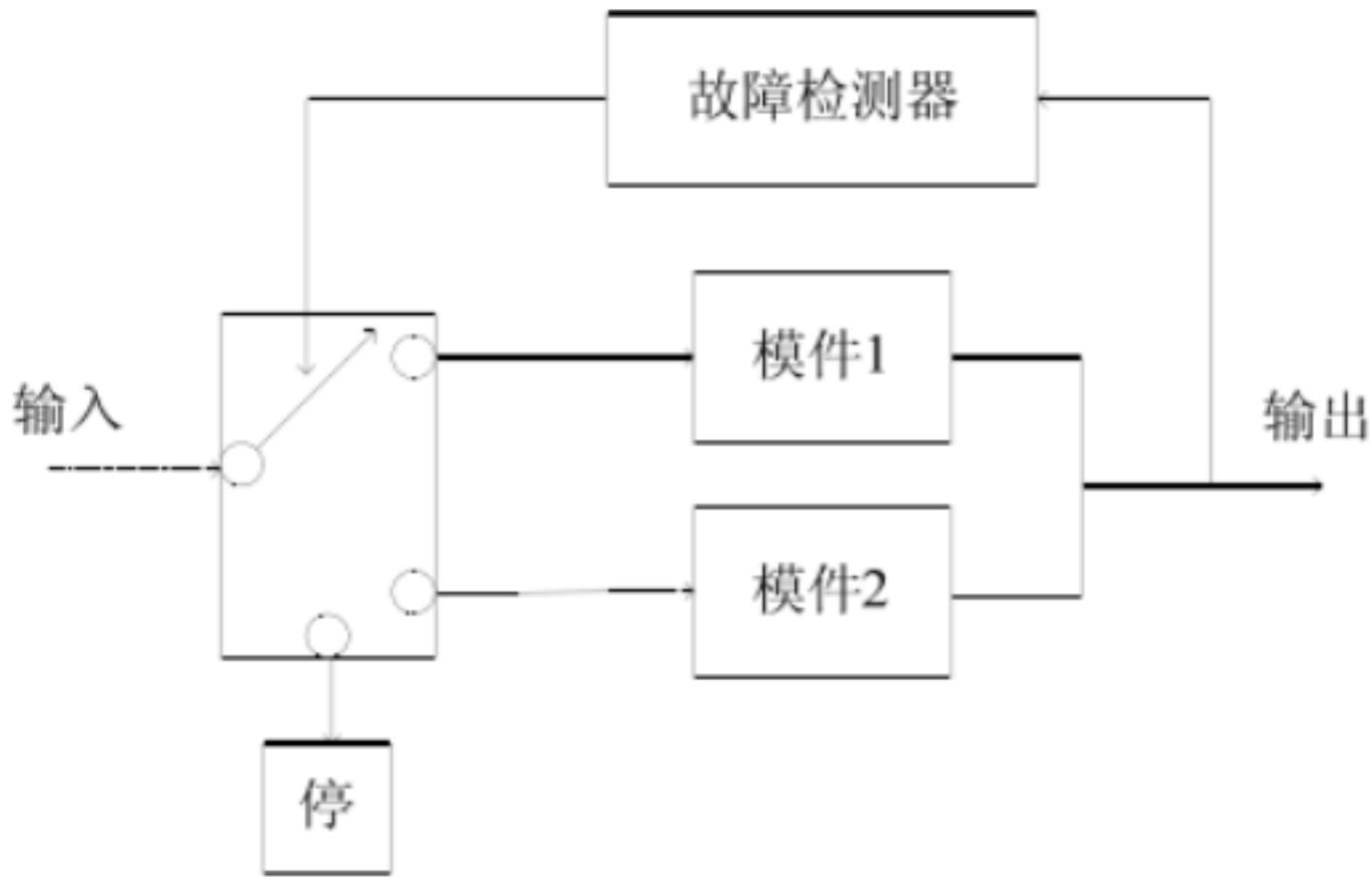

图 1 二冗余结构原理图

1.1.2 硬件智能容错方式的分类

硬件智能容错按其工作方式可以分为：静态冗余、动态冗余和混合冗余。

静态冗余容错是通过表决和比较屏蔽系统中出现的故障，如图 2 所示：

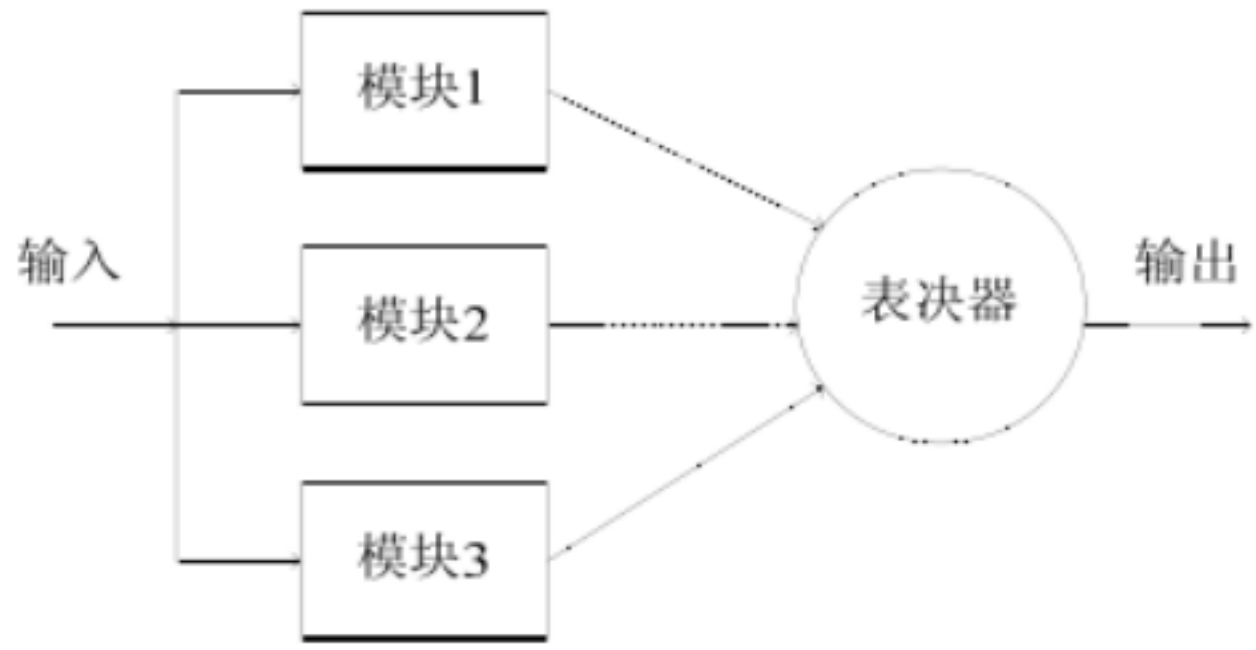

图 2 三模冗余（静态冗余） TMR 系统结构图

静态冗余容错的主要特点是：

（1）由于故障被屏蔽，所以不需要识别故障；

（2）容易与无冗余系统进行转换；

（3）所有模件都消耗能量。

动态冗余 的主要方式是多重模块相继运行来维持设备正常工作。当检测到工作模块出现故障时， 一个备用模块立即接替故障模块并投入工作。

动态冗余容错控制的主要特点是：

（1）仅有一个模件消耗能量；

（2）模件数目可随任务而改变，不会影响系统工作；

（3）转换装置和检测装置中任一故障都会导致系统失效。

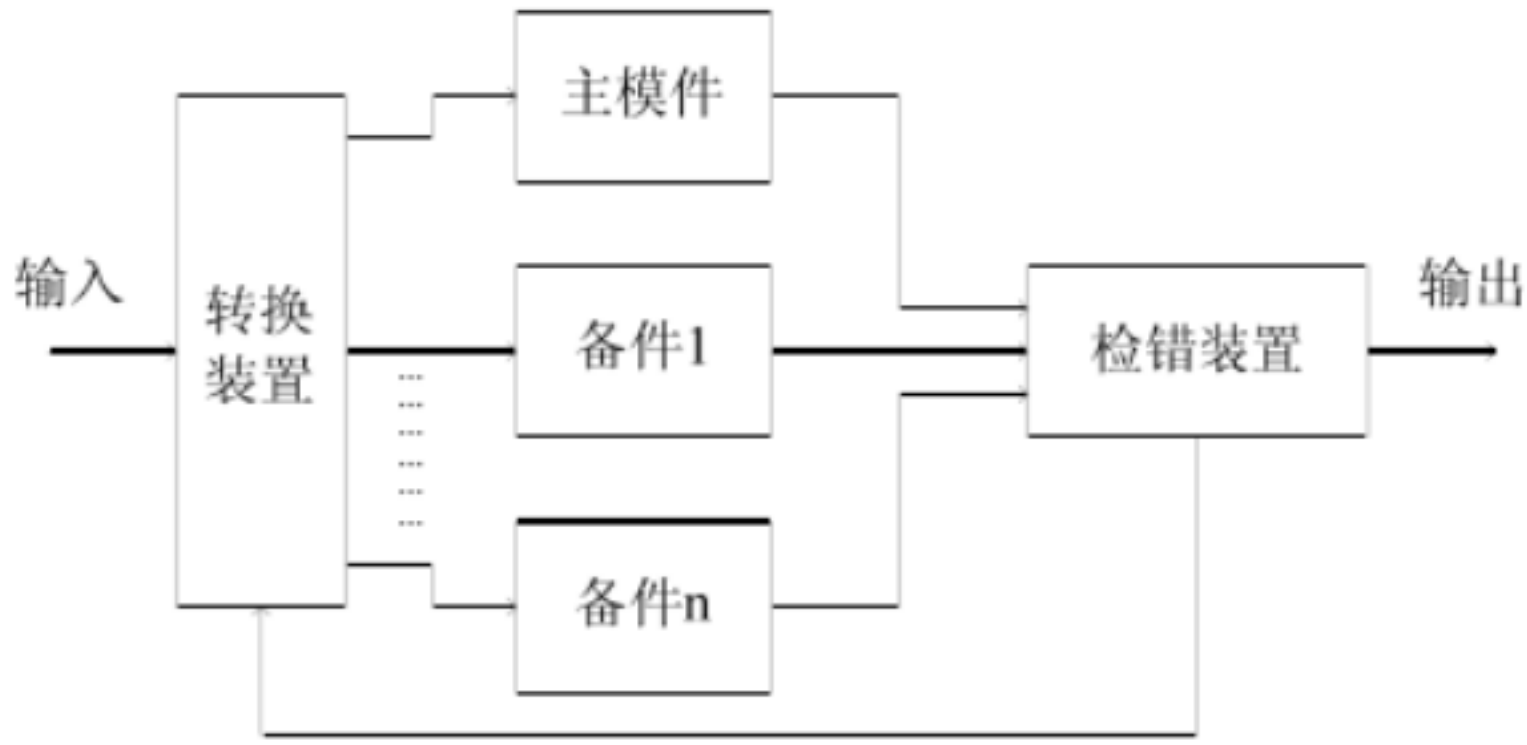

图 3 动态冗余容错控制结构图

混合冗余兼动态冗余和静态冗余之所长，通常用 H（n,k）来表示，如图 4 所示。图中的 V 为表决器， n 表示模块的总数， k 代表以表决方式实现静态冗余的模块数，而其余 N-K 个模块则作为表决系统中模块的备份。当参与表决的 k 个模块中（通常 k>=3）有一个模块出现故障时， 备份就替代该模块参与表决， 维持静态冗余系统的完整。当所有备份都被替换完后，系统就成为一般的表决系统。

如在硬件构成的逻辑系统中表决器是由开关电路实现的，而软件中表决需要通过软件断言 SA(Software Assertions)来实现。软件断言就是当软件在宿主系统中运行时， 对其进程或功能的正确与否做出判断的条件。

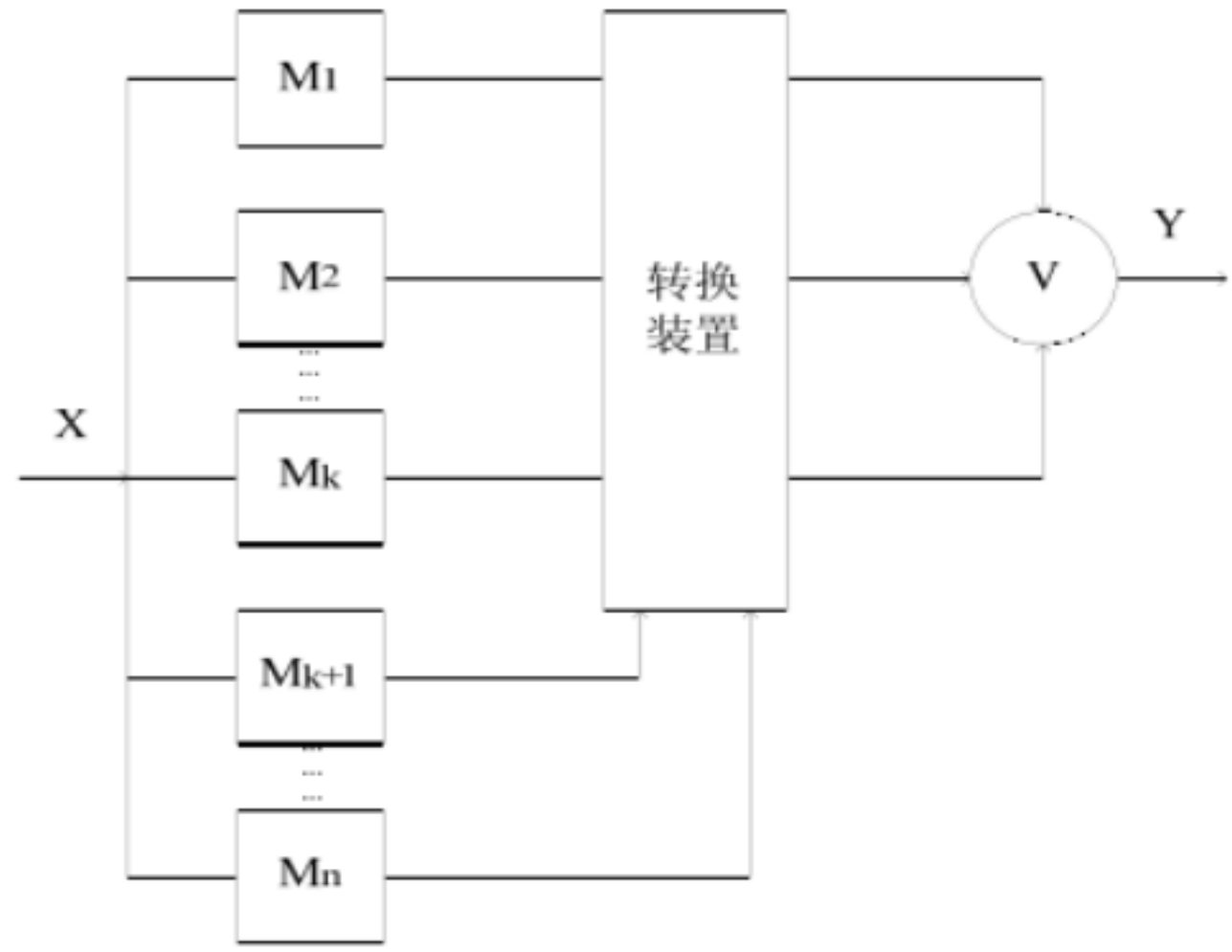

图 4　H(n,k) 系统结构

1.1.3 智能容错的故障处理方式

智能容错技术是一种外延广博的综合性技术。为了消除故障的影响，可以采用以下处理方式来实现　：

（1）故障检测

通过故障检测可以迅速准确地对故障进行定位。故障检测是容错的基础。故障检测方式可分两种：脱机检测，即进行检测时系统不能做有用的工作，联机检测，即检测与系统工作同步进行，它具有实时检测的能力。

（2）故障定位

在给定的故障条件下，找出故障原因，确定发生故障元件的具体位置。定位的详细程度视具体问题而定，　一般定位到进行系统重构所需的最小单元。

（3）故障屏蔽

故障屏蔽能够把故障效应掩盖起来，以防止故障对输出产生影响。故障屏蔽只能容忍故障，而不能给出故障警告，当冗余资源耗尽时，将使设备产生错误输出。

常用的故障屏蔽方法有多模表决冗余和屏蔽逻辑两种，多模表决冗余就是在设备的多个装置中，　只要至少有一个装置正常工作，　系统就能完成其功能；屏蔽逻辑主要用于门级电路的故障屏蔽，　它能有效地限制逻辑线路门输出的临界故障与亚临界故障。

（4）故障限制

故障限制就是规定故障的传播范围，把故障效应的传播限制到

某一区域内。故障限制可以用软件和硬件来实现。

（5）故障隔离

故障隔离就是将故障隔离起来以防其进一步扩散和对设备产生影响。

（6）故障修复

当设备发生故障经检测和定位后，就可采取更换、修理、自修复等方式使设备复原。

（7）系统重组

当设备发生故障时，通过任务的重新分配或内部器件的重新组合，以切除或替换故障部件。

（8）系统重构

重构就是把修复的模件重新加入到系统中去。

（9）系统恢复

系统恢复就是经过屏蔽　,重组等 ,使故障恢复到故障前的工作状态，不丢失或少丢失信息，　并保证下一步的正常运行，系统恢复通常用软件实现。

1.1.4 智能容错的实现方法

智能容错的实现方法分为 ：（1）故障信号检测；（2）故障特征识别；（3）故障状态预测；（4）故障维修决策；（5）故障容错控制。

故障容错的目的在于针对不同的故障源和故障特征，采取相应的容错处理措施，对故障进行补偿、消除或自动修复，以保证设备继续安全可靠运行， 或以牺牲性能损失为代价， 保证设备在规定时间内完成其基本功能。结构框图如图 5 所示。

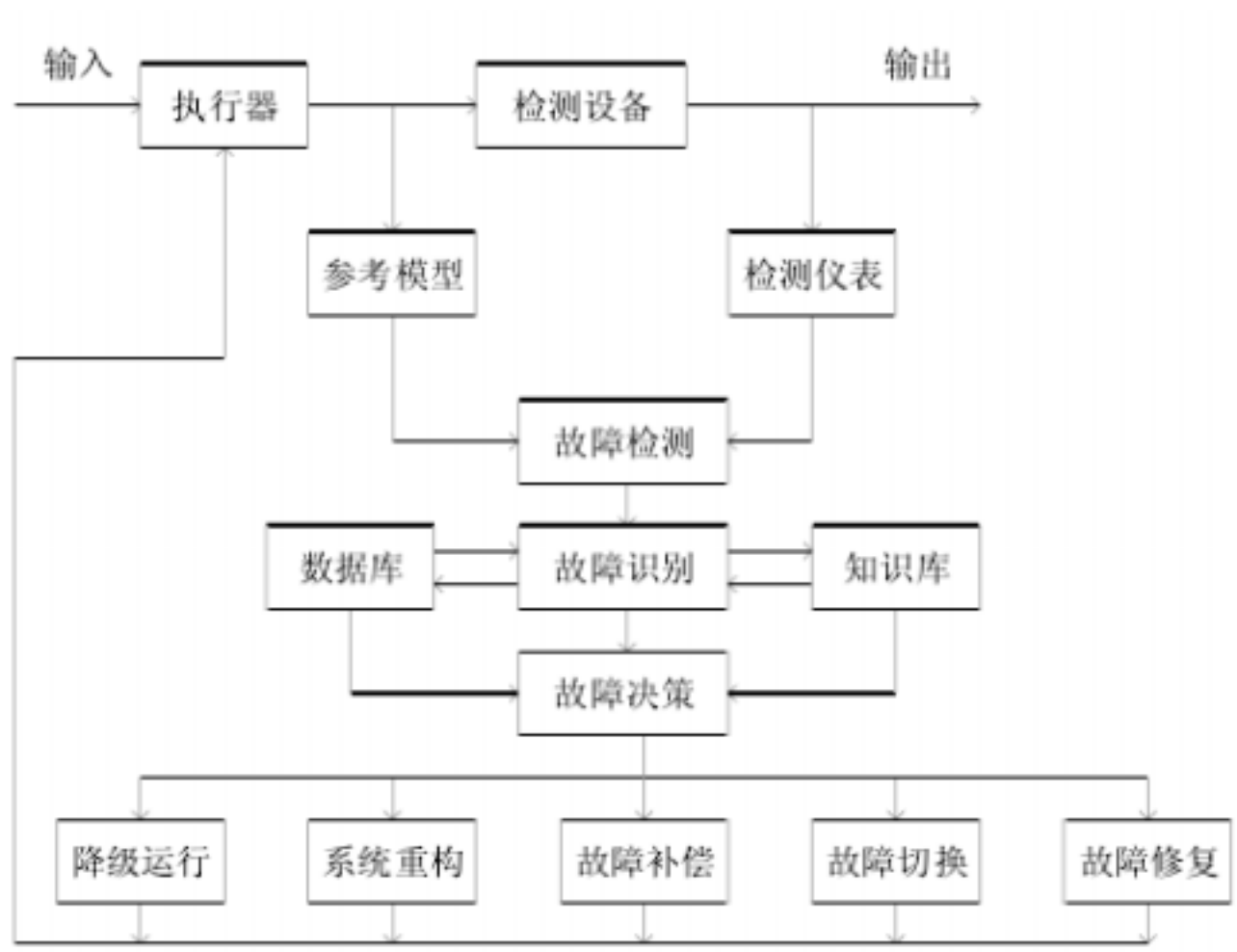

图 5 故障容错控制过程框图

2 冗余技术

所谓冗余（ Redundancy）就是多余资源，冗余技术可供用来处理故障，冗余技术分为 ：

（1）硬件冗余法，硬件冗余 HR（Hardware Redundancy）就是依靠附加硬件的冗余性和互补性来实现故障容错， 附加硬件通常采用储备形式， 当设备某个或某些关键部件发生故障后， 可以用备份硬件替代故障部件，以削弱或消除故障的影响。

（2）软件冗余法， 软件冗余 SR(Software Redundancy)可以通过增加软件功能来实现，

其中包括修改容错控制策略、重新配置系统软件、有效地降低设备的运行速度、多模块并行诊断决策等。

冗余附加技术指为实现上述荣誉另外所需的资源和技术，包括程序、指令、数据以及存放和调动他们的空间和通道。他们和硬件冗余中冗余备份一样， 在没有容错要求的系统中是不需要的， 而在容错系统中却是必不可少的。 以屏蔽硬件故障为目的容错技术中， 冗余附加技术包括：（1）关键程序和数据的荣誉存储和调用； （2）进行检测、表决、切换、重构、纠错、复算的实现。在屏蔽软件故障的容错系统中，冗余附加件的构成不同。

冗余附加件包括：（1）独立设计的相同功能冗余备份程序的存储及调用；（2）实现纠错误检测及恢复的程序； （3）为实现容错软件所需固化了的程序。 冗余、容错技术发展现状伴随着大规模和超大规模集成电路的发展， 硬件可靠性大大提高而价格却大幅度降低， 使采

用各种容错技术在经济上更易接受。

容错技术应用范围扩展于银行事务处理及各种实时控制系统，甚至许多通用计算机系统也采用了容错技术。 在七八十年代， 容错技术应用已经很广泛，例如 :1975 年的美国贝尔实验室的 3A 号 ESS 处理系统和美国 TANDEM16 容错事务处理系统： 1976 年的美国 AMDAHL470V/6 容错通用计算机和 1978 年容错空间计算机 FTSC；1979 年 BIM 推出容错的 4300 通用计算机系列； 1980 年容错多处理机 FTMP 及软件实现的容错计算机 sift 研制成功等等。

随着电子交易的日益广泛，出现了商用容错计算机市场和以分布式为体系的容错计算机系统。容错的 VLSI 技术和人工智能在容错技术上的应用——计算机故障诊断专家系统， 给冗余、容错技术的发展增添了新的活力。

冗余、容错技术理论的研究， 也是相当活跃的。 1952 年，冯·诺依曼作了一系列关于用重复逻辑模块改善系统可靠性的报告； 1956 年，他发表了论文《概率逻辑及用不可靠元件设计可靠的结构》 。

1971 年以来， IEEE 计算机学会容错技术委员会每年召开一次国际容错计算学术会议； 1987 年中国计算机学会成立了容错计算专业委员会等等。基于容错控制 (TFC)基本思想， FTC 的研究主要有被动容错控制 (Passives)和主动容错控制 (Activate)两种途径。

主动容错控制是在控制系统故障检测与 FDD(Fau1t Detection and Diagnose)的基础上，当 FDD 环节检测出系统故障后，重新调整控制器参数，甚至改变控制器结构，在保证系统稳定的前提下，尽量

恢复系统故障前的性能。 被动容错控制是设计对故障具有容忍能力的强鲁棒控制器， 被动容错控制的研究可以充分利用鲁棒控制技术的研究成果，且不受 FDD 发展水平的限制， 所以对于被动容错控制的研究取得的成果较多。

2. 1 主动容错控制

主动容错控制一般需要两个基本步骤 :控制系统的故障检测、诊断与隔离及控制系统重构。 控制系统故障检测、 诊断与隔离是在现代控制理论、可靠性理论、数理统计、信号处理技术、模式识别技术，以及人工智能和计算机控制技术等技术基础上的一门应用型的边缘学科， FDD 技术是容错控制重要的支撑技术之一。

由于控制系统 FDD 问题本身的复杂性和相关领域技术水平的限制， 虽然人们对它的研究已达到了一定的水平， 但至今仍没有解决这一问题特别有效的方法。

目前的控制系统 FDD 研究主要是沿着基于模型和基于知识两种途径展开。控制系统重构的方法主要有 :控制律重新调度、控制器重构设计和模型跟踪重组控制， 控制律重新调度的基本思想是， 在离线的情况下计算出各种故障条件下所需的控制律增益参数， 存储在计算机中，系统根据 FDD 单元所给出的结果，选择合适增益参数，实现对各种故障的容错控制 ;控制器重构设计是根据故障系统的新环境， 重新设置系统的工作点，并给出可改善系统性能的新控制器， 现有的控制器重构方法主要有基于直接状态反馈或输出反馈的方法， 以及基于动态补偿器的设计方法等 ;模型跟踪重组控制的基本原理是采

用模型参考自适应控制的思想，　使得被控过程的输出自适应地跟踪参考模型的输出，因此这种容错控制不需要　　FDD 单元。在主动容错控制方法中，能够较好地将　FDD 环节与系统重构相结合的是基于人工智能的容错控制方法。 在容错控制中所运用的人工智能方法主要是人工神经元网络　ANN(Artifictial Neural Network)　, 利用　ANN　对非线性特性的任意逼近能力和　ANN 所具有的从样本中学习、归纳和推理的能力，通过训练，使　ANN 能准确地估计出故障的大小，在此基础上再通过故障补偿来实现主动容错控制。被动容错控制

在目前的容错控制研究中，因为不受控制系统　　DFD　环节的限制，被动容错控制相对于主动容错控制要更容易实现，　　己有的可以实现被动容错控制的主要方法有　：

完整性控制器设计、　同时镇定和可靠镇定。　完整性控制 (Intgearl Contron)的概念由　Niederlinski　在　1971 年提出，　完整性控制是研究最早的一种容错控制技术，因为控制系统中传感器和执行器是最容易发生故障的部件，所以完整性控制具有很高的应用价值，在控制理论中，称多变量系统中出现故障时仍能保持系统稳定性的控制器为完整性控制器，　完整性控制器设计是多变量系统中特有的问题　;多模型设计方法又称同时镇定问题，自从　　Ackermann，　Sakes 和 Vidyasagar 等人提出来以后，己经成为容错控制的一个重要研究方向，同时稳定容错控制的设计方法是力求寻找一个公共的状态反馈控制器，使之能够同时稳定尽可能多的故障情况下的系统模型，同时兼顾到系统的动、静态品质特性的要求　;使用多个补偿器进行可靠

镇定的概念是由 Silage 于 1980 年最早提出，可靠镇定实际上是关于控制器的容错问题。

与被动容错控制相比较，主动容错控制具有更多的优点。从理论上讲，被动容错控制是故障情况下的强鲁棒控制，主动容错控制是故障情况下的强自适应控制。被动容错控制即使在系统正常的情况下控制律也要满足故障条件下的要求，这在系统正常时显然是一种过高的要求，设计未免过于保守，必然要以牺牲性能指标为代价。另外，在预想故障数目较多时，被动容错控制问题可能根本没有解，所以被动容错控制有较大的局限性。

基于控制系统 FDD 的主动容错控制实质是一种强自适应控制，它通过实时地对系统进行故障检测与诊断，当检测出系统故障后，根据不同的故障采取相应的措施，保证系统的稳定性和维持一定的性能指标。主动容错控制所用的主要方法是控制律重构和故障补偿，前者需要根据故障重新设计控制器，后者则是利用故障的信息确定一个控制补偿量，目的都是力图使故障后的系统尽量接近甚至等价于原系统。对于演变速度较慢的所谓软故障，多模自适应方法比较适合，但多模自适应方法中存在较多的算法上的问题，限制了这种方法的使用。

2.2 容错控制研究中需要解决的主要问题

尽管控制系统 FDD 和 TFC 技术的研究在理论上已取得了较为丰富的成果，但距离实际工程应用的要求还有相当大的差距，理论上也还有许多问题有待人们去研究和探索。

本文主要研究了实时系统多机冗余、容错系统的故障检测与诊断、控制系统重构、 容错实时运行库技术以及容错控制在工程中的应用等问题，而在目前的研究中，上述领域主要存在的问题分述如下 ：

（1）控制系统故障检测与诊断中存在的问题： 控制系统故障的模型是从理论上进行 FDD 研究的前提，但现有故障建模方法简单，与实际系统故障具有的复杂性和多样性成为一对急待解决的矛盾。 目前还没有一个在 FDD 和 FTC 中比较统一的故障表示方法。就拿 CPU 测试来说，多数的结构测试法需要详细的系统逻辑电路图，并在此基础上建立故障模型。

结构测试法通过验证电路中没有任何符合故障模型的故障存在，来说明电路中没有影响电路功能的实际物理故障存在。 由于微处理器是一个输入、 输出引线数目有限， 内部结构异常复杂的大规模集成电路，很难在逻辑门这一级确立准确的故障模型 ;另一方面受到引出脚数目的限制，使得故障的可控制性与可观察性大大降低 ;此外控制逻辑部分与数据处理逻辑部分都在一块芯片上， 不能预先假定哪一部分总是好的。

（2）系统重构方面存在的主要问题： 目前的系统重构问题研究较少。现有的运用广义逆的方法、基于状态反馈或输出反馈、特征值和结构配置等方法， 仅是从数学模型角度将系统恢复， 而不是从系统性能角度恢复， 所以重构后的系统鲁棒性不能保证， 有时甚至稳定性也难以保证。而且，多数重构对系统模型的要求也很苛刻。就目前的控制系统 FDD 研究水平来看，想获得故障后系统模型的全部信息是

相当困难的， 所以寻求故障后控制律完全重构是理想化的。 由于故障的大小是未知的， 所以在设计时进行稳定性分析是非常困难的， 只能对预想的故障进行稳定性分析。

（3）实时系统和容错技术相结合存在的主要问题： 如何将软件容错技术有机地融合到实时系统中去， 具有与发展软件容错技术本身同样重要的地位。

目前，尽管实时系统的软件容错技术已经取得了非常巨大的进步，但在实时系统中并未得到充分地应用。

（4）用户在冗余、容错系统设计中存在的主要问题：应用程序设计者在考虑如何实现应用软件的功能要求同时，要兼顾软件容错，这不可避免地， 甚至是成倍地加大了应用系统开发的工作量， 增加系统的复杂程度， 反而加大应用出错的可能性。 应用层容错将容错机制的实现和应用程序融为一体， 当需要在同一操作系统上开发新的应用时，所有工作必须从头开始，不符合工程实际中提高软件重用性，开发低成本、高可靠系统的大趋势。

（5）容错控制理论运用于工程实际时需要解决的问题： 容错控制在理论研究上比较困难， 在实际工程中的应用更少。 由于不同的工程领域所遇到的问题有很大的差异， 不可能以一个统一的框架来解决所有的问题。理论研究所用的模型和假设同工程实际的差别比较大，这也是容错控制理论在工程实际中运用所遇到的主要困难。 各个领域的工程技术人员，应结合自己的工程实际，选择相应的控制方案。因此，统一系统架构，构建支持多种主流冗余、容错模式的运行库，对

工程设计人员来说是很重要的。