

摘 要

本文在详述传统驱动桥设计方法及其特点基础上,详细研究了微型汽车驱动桥智能化设计的实现策略,并构造了基于 I-DEAS 的微型汽车驱动桥智能化设计软件系统。

全文以软件工程思想为基础,从概要设计、详细设计、几何建模及分析、系统的集成方法四部分阐述了微型汽车驱动桥智能化设计系统的设计思路,最后给出了完整的运行实例。文中对设计参数的层次化定义、约束的分类及其动态化和智能化处理提出了有效的解决方案,并建立了参数信息数据库、尺寸约束函数库和知识约束库。文中还通过对参数约束数学模型的分析并结合驱动桥工程化设计过程,设计出搜索算法和参数更新算法。算法通过调用数据库、函数库和知识库,构成了微型汽车驱动桥智能化设计系统的核心。所形成的软件系统达到了原设计指标要求,并已经过实例测试。

关键字: 驱动桥, 参数化, 智能化, 知识模型, 约束

Abstract

After specifying the characteristic of the traditional driving axle's design method, the implementation strategy of the minitype automobile driving axle is studied by intelligentize method, and a software system for minitype automobile driving axle based on I-DEAS is established in this dissertation.

Based on the software engineering idea, the design ideas for this system are expanded in four parts: concept design, detail design, geometrical modeling and analyzing, the system's integration. Effective approaches are put forward which to define hierarchical levels of design parameters, classify along with dynamic and intelligent solve various engineering design constraints, and parameters information database, dimension constraint functions database and knowledge constraint database are establishes. By analysing the mathematical model and the design process of driving axle, the search algorithm and the parameter update algorithm are given in this dissertation. These algorithms can call the three databases, and this is the kernel of the system.

Keywords: driving axle, parametric, intelligentize, knowledge model, constraint

1 绪论

1.1 论文的选题背景

本课题是四川建安工业有限公司为了提高驱动桥的开发效率、缩短设计周期而提出的。四川建安工业有限公司是一家以汽车零部件为主要产品的生产企业，其中轻、微轿车前、后桥已达到年产 40 万台套的生产能力，成为国内最大的微车驱动桥生产企业。

通过在四川建安工业有限公司两个月的调研实习，发现他们要设计一种新的驱动桥总是按照例行的步骤去完成的，包括计算、校核，设计出各零件，画出二维图，试验，反复修改等等，最后得出一种比较完善的设计方案。他们在设计时经常参考以前设计出的驱动桥并稍加修改（主要是尺寸的变化），最终形成系列化的驱动桥。而这些例行的工作却还是由设计者去一步一步的手工完成，将时间都花在这些没有创新的枯燥的事上；更让人头痛的是，人工完成这些工作将会使开发周期变得漫长。

另外，建安工业有限公司在设计驱动桥时全部采用传统的二维制图。根据二维图纸去加工实物时经常发现结构上有许多不合理之处，并重新修正，浪费了时间、人力、物力、财力，要提高效率最好的办法是直接设计出三维实体图，并进行各种模拟分析，达到省时、省力、省钱的目的。而实现这种功能的软件也相当多，如：I-DEAS, UG, PRO/E, SolidWork 等等。

再看看国外的情况吧！国外一些汽车零部件生产企业已经用参数化设计技术研制出一些实用的产品开发软件系统（如英国 Locus 集团采用的制动器设计系统），使制动器的设计周期大大缩短。各种软件的开发，使得汽车零部件的质量和性能得到很大地改善，并大幅度地节省了开发时间和成本。虽然我们还不能对他们的产品有更多的了解，但至少说明开发这种产品的实用性、可行性和迫切性。

1.2 关于微型汽车驱动桥设计方法

1.2.1 微型汽车驱动桥的功用及设计要求

驱动桥的功用是将传动轴或变速器传来的转矩增大并适当降低转速后分配给左、右驱动车轮。驱动桥组成包括主减速器、差速器、半轴、驱动桥桥壳等^[1]。

驱动桥是汽车传动系中主要总成之一。驱动桥的设计是否合理直接关系到汽车使用性能的好坏。因此，设计中要保证^[1]：

- (1) 所选择的主减速比应保证汽车在给定使用条件下有最佳的动力性和燃料经济性；
- (2) 当左、右两车轮的附着系数不同时，驱动桥必须能合理的解决左、右车轮的转矩分配问题，以充分利用汽车的牵引力；

- (3) 具有必要的离地间隙以满足通过性的要求;
- (4) 驱动桥的各零部件在满足足够的强度和刚度的条件下, 应力求做到质量轻, 特别是应尽可能做到非簧载质量, 以改善汽车的行使平顺性;
- (5) 能承受和传递作用于驱动车轮上的各种力和转矩;
- (6) 齿轮及其他传动部件应工作平稳, 噪声小;
- (7) 对传动件应进行良好的润滑, 传动效率要高;
- (8) 结构简单, 拆装调整方便;
- (9) 设计中应尽量满足“三化”(产品系列化、零部件通用化、零件设计的标准化)的要求。

1.2.2 微型汽车驱动桥的传统设计步骤

驱动桥传统设计方法和步骤包括:

- (1) 驱动桥零部件选型及主要受力件的设计;
 - ① 选择驱动桥结构型式, 主要是断开式和非断开式驱动桥的选择;
 - ② 选择主减速器结构型式, 微型汽车驱动桥大多数采用单级主减速器;
 - ③ 选择主减速器锥齿轮的支承型式, 主动锥齿轮主要是悬臂式和骑马式两种;
 - ④ 计算选择主减速器齿轮的主要参数;
 - ⑤ 计算主减速器齿轮的几何参数;
 - ⑥ 主减速器齿轮的强度校核;
 - ⑦ 计算选择主减速器的轴承;
 - ⑧ 选择差速器齿轮主要参数;
 - ⑨ 计算差速器齿轮几何参数;
 - ⑩ 差速器锥齿轮的强度校核;
- (2) 根据以上的设计计算画出各零部件的三视图;
- (3) 对样件进行各种试验, 以确定设计是否合理;
- (4) 若不合理, 反馈给设计人员改进设计;
- (5) 投入生产。

1.3 汽车及零部件 CAD 的研究现状

计算机辅助设计 (CAD: Computer Aided Design) 技术一般泛指以计算机作为辅助工具的各种应用技术^[2]; 是以计算机、外围设备及其系统软件为基础, 包括二维绘图设计、三维几何造型设计、有限元分析 (FEA) 及优化设计、数控加工编程 (NCP)、仿真模拟及产品数据管理等内容。随着 Internet/Intranet 网络和并行、高性能计算及事务处理的普及, 异地、协同、虚拟设计及实时仿真也得到广泛应用。它是计算机系统在工程和产品设计的各个阶段和过程中, 为设计人员提供快速、有效的工具和手

段，以达到最佳的设计效果的一种技术。

在 CAD 软件发展初期，CAD 的含义仅仅是图板的替代品，即：意指 Computer Aided Drawing(or Drafting)，而非现在我们经常讨论的 CAD(Computer Aided Design)所包含的全部内容。CAD 技术以二维绘图为主要目标的算法一直持续到 70 年代末期，以后作为 CAD 技术的一个分支而相对单独、平稳地发展。早期应用较为广泛的是 CADAM 软件，近十年来占据绘图市场主导地位的是 Autodesk 公司的 AutoCAD 软件。在今天中国的 CAD 用户特别是初期 CAD 用户中，二维绘图仍然占有相当大的比重。

CAD 技术的发展大致可以分为四个阶段^[3]：

- (1) 蓬勃发展和进入应用时期（60 年代）
- (2) 广泛使用的时期（70 年代）
- (3) 突飞猛进的时期（80 年代）
- (4) 开放式、标准化、集成化和智能化的发展时期（90 年代）

目前 CAD 系统的发展趋势主要体现在以下几方面^[4]：

- (1) CAD 系统应用面向产品的整个生命周期
- (2) CAD 系统应充分考虑产品信息的继承性
- (3) CAD 系统应满足并行设计的要求
- (4) CAD 系统与产品信息标准化相结合
- (5) 产品模型的可转换性特别是二维和三维产品模型间的互相转换可满足不同的需要
- (6) 面向全国乃至全球的产品信息编码系统
- (7) 软件系统要具有很好的可移植性和自组织性
- (8) 智能化
- (9) 虚拟现实技术

汽车工业是最先应用 CAD 技术的行业之一，目前在发达国家的汽车行业中，CAD 技术已得到了广泛的应用，并取得了巨大的经济效益，新的轿车产品的开发周期已缩短至 2~3 年，汽车 CAD 系统的软硬件都已达到较高水平，并仍在向智能化、集成化和标准化的方向发展^[5]。

早在 80 年代初，福特公司已着手 CAD 系统规划，建成了以工作站为主体的环行网络系统，1985 年已经有一半以上的产品设计工作使用图形终端实现，1986 年新开发的 TAURUS 和 SABLE 轿车，大约 70%的钣金件采用 CAD/CAE 设计，到 90 年代初已全面实行产品开发的 CAD/CAE，应用达 100%。福特公司 1990 年工作站已达 2000 台，以 SGI 工作站（约占 70%）和 CV 工作站（约占 18%）为主，其应用软件主要为自行开发的 PDS 和 CAD/CAE。1993 年以后，福特汽车公司提出了 C3P (CAD/CAM/CAE/PDM)概念，并决定今后将采用 I-DEAS 软件作为其主流核心软件。

美国通用汽车公司应用 CAD 技术的情况与福特类似, 其软件主要为自行开发的 CGS (Corporate Graphics System) 和 Unigraphics Solutions 公司的 Unigraphics 系统, 这两种软件系统构成公司的三维 CAD 基础平台。通用公司的下一步目标是 100% 采用 CAD 技术, 取消中间过程, 将计算机与加工中心直接连接, 最终使系统网络化。

德国各大汽车公司普遍采用 CATIA 作为其 CAD 系统的主导软件。1994 年, 德国大众集团决定用 CATIA 和 Pro/Engineer 作为其将来开发新车型的主导 CAD 系统。

目前各大汽车公司的 CAD 系统还与 CAE/CAM 相结合, 并进一步集成为 CIMS (计算机集成制造系统)。CIMS 已成为汽车工业现代化生产制造的发展方向和主要标志之一。

1.4 关于参数化设计

参数化设计 (Parametric) (也叫尺寸驱动 Dimension-Driven) 是指设计对象模型的尺寸用变量及其关系表示, 而不需要确定具体数值^[6], 是 CAD 技术在实际应用中提出的课题, 它不仅可使 CAD 系统具有交互式绘图功能, 还具有自动绘图的功能。目前它是 CAD 技术应用领域内的一个重要的、且待进一步研究的课题。利用参数化设计手段开发的专用产品设计系统, 可使设计人员从大量繁重而琐碎的绘图工作中解脱出来, 可以大大提高设计速度, 并减少信息的存储量。

1.4.1 参数化设计原理概述

参数化设计是建立在参数驱动机制之上的, 参数驱动机制是基于对图形数据的操作。通过参数驱动机制, 可以对图形的几何数据进行局部修改, 实现结构和形态的变化 (见图 1.1)。若在修改参数的同时还要满足图形的约束条件, 还需要有使尺寸根据约束关系实现变动的机制, 即约束联动。一个图形的约束可能十分复杂, 而且数量很大, 而实际由用户控制的, 即能够独立变化的参数一般只有几个, 称之为主参数或主约束, 其他约束可由图形结构特征确定或与主约束有确定关系, 称之为次约束^[7]。对主约束不能简化, 对次约束可以进行简化, 并有图形特征联动和相关参数联动两种方式。

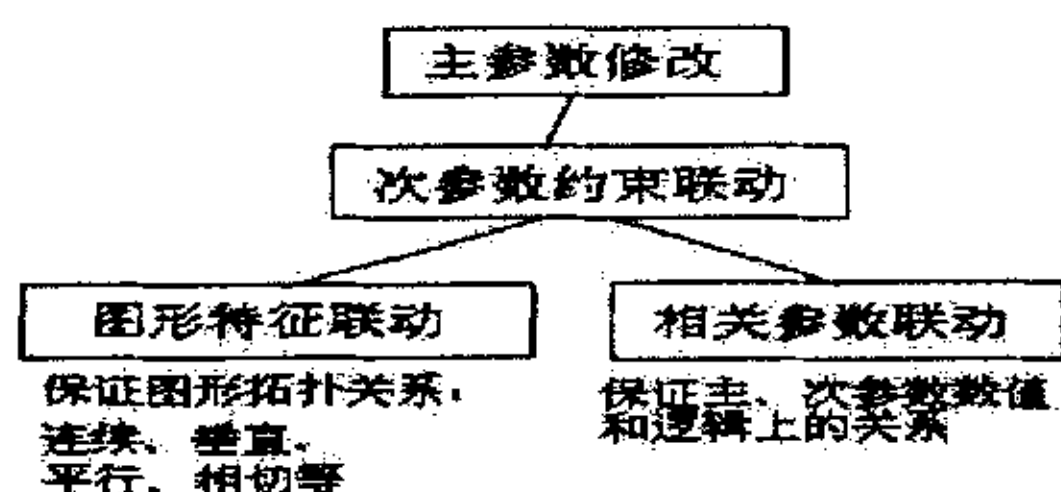


图 1.1 参数驱动机制示意图

所谓图形特征联动就是保证在图形拓扑关系不变的情况下, 对次约束的驱动, 亦

即保证连续、相切、垂直、平行等关系不变。反映到参数驱动过程就是要根据各种几何相关性准则去判别与驱动点(即主约束参数)有上述拓扑关系的实体及其几何数据,在保证原关系不变的前提下,求出新的几何数据,称这些几何数据为从动点。这样,从动点的约束就与驱动参数有了联系。依靠这一联系,从动点得到了驱动点的驱动。

所谓相关参数联动就是建立次约束与主约束在数值上和逻辑上的关系。在参数驱动过程中,始终保持这种关系不变。相关参数的联动方法使某些不能用拓扑关系判断的从动点与驱动点建立了联系。使用这种方式时,常引入驱动树,以建立主动点、从动点之间的约束关系的树形结构,便于直观地判断图形的驱动与约束情况。

参数驱动是一种新的参数化方法,其基本特征是直接对数据库进行操作。因此它具有很好的交互性,用户可以利用绘图系统全部的交互功能修改图形及其属性,进而控制参数化的过程;与其他参数化方法相比较,参数驱动方法具有简单、方便、易开发和使用的特点,能够在现有的绘图系统基础上进行二次开发。

1.4.2 基于特征的参数化造型思想

目前在 CAD 应用中,参数化设计应用越来越广,因此就出现了将参数化设计应用到特征设计中去,使得特征具有可调整性,主要是针对特征的几何和拓扑信息。利用混合法来建立特征模型,并将参数化方法引入到特征造型中去,使形状特征可以根据需求调整变化,这就是基于特征的参数化设计^[8]。

随着计算机的普及,人工智能的应用范围越来越广泛,但在 CAD 方面的应用却碰到些问题。造成这种状况的原因主要来自于对实体的描述和表示上,采用何种描述方法才能使得计算机很好的理解实体,以进行合理有效的几何推理成为人们目前聚焦的问题。传统的实体表示方法使用简单的原始的几何元素来表达实体,比如线条、圆弧、圆柱以及圆锥等,这样显得很枯燥、单调,计算机很难识别和理解这样粗糙的模型。因此就迫切需要发展一种建立高层次实体的基础上的实体表示法,这种实体需要包含更多的工程信息,这种实体就被称为特征。并且由此提出了以特征为基础的特征造型的基于特征的设计方法。目前特征造型在一些主流的 CAD 系统中有使用,但仍然处于研究阶段。

基于特征的参数化设计将基于特征的设计与参数化设计有机的结合起来,使用较完整的带有语义的特征描述方式,使特征本身就包含参数化变动所需的成员变量和成员函数,将面向对象的技术应用于特征的描述,在造型中也使用参数化,随时可调整产品结构,尺寸,并因此带动特征自身的变动,实现产品的基于特征的参数化设计。由于不同的行业对特征类型需求不同,所以目前的特征设计的思想应用受到一定的限制,特征仍处于研究阶段。

1.4.3 参数化设计方法应用现状

自从 PTC 公司于 1988 年推出参数化造型系统 Pro/E 以来,各大软件公司纷纷推

出自己的基于参数化造型模块,他们分别采用了不同的技术,各有所长^[9]。

(1) Unigraphics(UG)

UG 是 Unigraphics Solutions 公司的拳头产品。在 UG 中,优越的参数化和变量化技术与传统的实体、线框和表面功能结合在一起,这一结合被实践证明是强有力的,并被大多数 CAD/CAM 软件厂商所采用。

UG 最早应用于美国麦道飞机公司。90 年代初,美国通用汽车公司选中 UG 作为全公司的 CAD/CAE/CAM/CIM 主导系统,这进一步推动了 UG 的发展。1997 年 10 月 Unigraphics Solutions 公司与 Intergraph 公司签约,合并了后者的机械 CAD 产品,将微机版的 SOLIDEDGE 软件统一到 Parasolid 平台上。由此形成了一个从低端到高端,兼有 Unix 工作站版和 WindowsNT 微机版的较完善的企业级 CAD/CAE/CAM/PDM 集成系统。

(2) I-DEAS

I-DEAS 是美国 SDRC 公司开发的 CAD/CAM 软件。该公司是国际上著名的机械 CAD/CAE/CAM 公司,在全球范围享有盛誉。该件是高度集成化的 CAD/CAE/CAM 软件系统。它帮助工程师以极高的效率,在单一数字模型中完成从产品设计、仿真分析、测试直至数控加工的产品研发全过程^[10]。I-DEAS 是全世界制造业用户广泛应用的大型 CAD/CAE/CAM 软件。

I-DEAS 在 CAD/CAE 一体化技术方面一直雄居世界榜首,软件内含诸如结构分析、热力分析、优化设计、耐久性分析等真正提高产品性能的高级分析功能。

(3) Pro/Engineer

Pro/Engineer 系统是美国参数技术公司(Parametric Technology Corporation, 简称 PTC)的产品^[12]。Pro/Engineer 系统主要功能如下:

- ① 真正的全相关性,任何地方的修改都会自动反映到所有相关地方。
- ② 具有真正管理并发进程、实现并行工程的能力。
- ③ 具有强大的装配功能,能够始终保持设计者的设计意图。
- ④ 容易使用,可以极大地提高设计效率。

Pro/Engineer 系统用户界面简洁,概念清晰,符合工程人员的设计思想与习惯。整个系统建立在统一的数据库上,具有完整而统一的模型。

(4) SolidWorks

SolidWorks 是生信国际有限公司推出的基于 Windows 的机械设计软件。SolidWorks 是微机版参数化特征造型软件的新秀,SolidWorks 是基于 Windows 平台的全参数化特征造型软件,它可以十分方便地实现复杂的三维零件实体造型、复杂装配和生成工程图。该软件可以应用于以规则几何形体为主的机械产品设计与生产准备工作中,其价位适中。

1.5 课题的研究内容和研究方法

论文主要的研究内容和研究方法包括：

(1) 参数化图库建立

① 对收集到的各种微型汽车驱动桥外形结构离散化。对所设计的零件进行分类，然后建立每种结构类型的结构形式模板，模板必须能够反映该系列现有常用零件中可能出现的结构。以几种典型的驱动桥外形为主模型，配合其它类型的驱动桥外形，对其离散化，提取典型特征。

② 根据给定的整车参数确定出几个重要尺寸。如果将零件模型所有的或大多数的结构尺寸都作为参数让用户输入，那么软件系统将没有多少实际价值。因为用户要输入这些尺寸必须首先设计出零件，而这一过程是我们想让计算机完成的。因此必须首先确定出由整车参数决定的少数几个关键尺寸，如齿轮、轴承等尺寸参数。

③ 由重要尺寸确定出其他所有尺寸。这一步工作量比较大，要考虑驱动桥的整个设计过程，要将许多设计经验融入其中。

(2) 接口设计

接口的主要作用是实现数据的转化，即将前处理的临时数据转换为格式化的数据文件，同时传给后续的图形处理。实现前处理与图形处理部分信息传递是通过一个独立接口模块来实现，接口模块的主要功能是产生格式化的接口文件。由图形处理部分对其进行处理。

(3) 图形处理

图形处理在 CAD/CAM/CAE 集成软件系统的内部实现。采用 CAD/CAM/CAE 集成软件系统的宏语言和其高级语言实现。

(4) 编写系统界面

用可视化高级语言编写微型汽车驱动桥智能化设计系统界面，将前面几步所做的联成一整体，完成用户数据输入、尺寸计算、调用图形库等功能，使用户能在此环境下完成整个设计过程。

系统界面包括以下几类：

① 整体参数输入界面。可以仅通过整体参数的输入完成整个驱动桥的设计，不用涉及单个零件的详细设计。其实，整个设计过程都由计算机自动完成。

② 单个零件参数输入界面。用户可以只设计某一个或某几个零件，而不用生成整个驱动桥。也可以通过对每一个零件的单独设计来完成整个驱动桥的设计。

③ 经验值选取界面。驱动桥设计过程中会涉及许多经验值的选取，系统会默认一个值，如果设计人员想修改，可以通过此模块来完成。

2 微型汽车驱动桥智能化设计系统概要设计

2.1 系统需求分析

在需求分析阶段，我们的焦点是“做什么”，而不是“怎么做”。不是具体地解决问题，而是准确地确定“为了解决这个问题，目标系统必须做什么”^[13]。主要是确定目标系统必须具备哪些功能。

用户了解他们所面对的问题，知道必须做什么，但是却不能完整、准确地表达自己的要求；软件开发人员知道怎样通过软件实现人们的要求，但是对特定用户的具体要求并不完全清楚。因此在需求分析阶段软件开发人员必须和用户密切配合，充分交流信息，以得出经过用户确认的系统逻辑模型。并与软件开发技术人员进行交流，以保证在目前现有的人员条件和技术条件下，用户需求是可实现的，系统逻辑模型是合理可行的。

2.1.1 软件系统的设计要求

我们首先是与用户交流，获取他们的各种需求，以下就是从用户处获取的需求信息：

- (1) 系统的最终输出是驱动桥的三维图形；
- (2) 进入系统后用户可以选择菜单开始一个新的驱动桥设计项目；
- (3) 用户可以对所要设计的驱动桥进行组件选型和总体选型设计，包括如下项目：
 - a. 选择驱动桥结构型式，主要是断开式和非断开式驱动桥的选择；
 - b. 从动锥齿轮支承型式，主要是悬臂式和骑马式；
 - c. 半轴型式，主要是半浮式，3/4 浮式和全浮式；
 - d. 主减速器齿轮类型，主要是双曲面齿轮和螺旋锥齿轮；
 - e. 主减速器减速型式，主要是单级减速器、双级减速器、双速减速器、单级贯通、双级贯通和单级轮边减速；
 - f. 桥壳结构型式，主要是三段可分式、二段可分式、整体式和组合式；
 - g. 主动锥齿轮支承型式，主要是悬臂式和骑马式；
 - h. 行星齿轮数目，主要有 2、3、4 三种数目；
 - i. 差速器结构型式，主要有对称式圆锥行星齿轮差速器和防滑差速器；
 - j. 主动齿轮远端轴承，主要有圆锥滚子轴承和深沟球轴承；
 - k. 主动齿轮近端轴承，主要有圆锥滚子轴承和深沟球轴承；
 - l. 差速器轴承，主要有圆锥滚子轴承和深沟球轴承；
 - m. 半轴轴承，主要有圆锥滚子轴承和深沟球轴承；
- (4) 用户可以以如下三种方式进行设计：

a. 从整车参数开始设计。这要求用户输入设计驱动桥所需的原始参数，系统由这些参数计算出关键参数，然后由关键参数推导出所有几何参数；

b. 从关键参数开始设计。相当于 a 中跳过计算关键参数这一步，然后由关键参数推导出所有几何参数；

c. 直接输入几何参数。相当于 b 中跳过由关键参数推导出所有几何参数这一步，几何参数直接由用户输入，然后生成相应的几何图形；

(5) 若用户采用的是从整车参数开始设计，则有原始参数输入的功能，它包括以下几部分参数输入界面：

a. 主减速器设计所需参数；

b. 差速器齿轮设计所需参数；

c. 半轴设计所需参数；

d. 桥壳设计所需参数；

e. 选择轴承所需参数。

(6) 系统有对所有输入的原始参数进行处理的功能。即通过这些参数，结合用户的选型信息，计算推导出所有计算几何尺寸参数所需的关键参数；

(7) 系统有对所有生成的关键参数进行处理的功能。即通过这些参数计算推导出所有几何尺寸参数；

(8) 有一个浮动工作区来显示用户生成的驱动桥的所有零部件，并可以通过点击某零件来查看此零件的几何三视图，此三视图标出了所有尺寸以方便用户查看尺寸；

(9) 用户可以随时修改零件的输入参数，因此有一个零件尺寸参数列表让用户输入修改值；

(10) 当用户修改完部分参数后必须有更新相关参数的功能，以保证相关参数被及时更新，避免数据匹配错误；

(11) 当用户完成设计后，可以从应用程序进入 I-DEAS 系统，根据用户设计完成的参数值形成驱动桥的三维模型；

(12) 若用户由直接输入几何参数方式设计，则无须原始参数的输入也无须计算，但可以让用户选择想要进行设计的零件；菜单中有一个添加删除零件功能使得用户能随时在工作区的列表中添加删除零件；

(13) 用户可以从菜单中进入 I-DEAS 系统对用户设计完的零部件进行有限元分析；

(14) 系统有对设计完的驱动桥进行设计评估的功能；

(15) 要求有联机帮助文档。

(16) 要求界面简单，操作方便

2.1.2 系统可行性分析

可行性分析的目的是用最小的代价确定在问题定义阶段所确定的系统的目标和规模是否现实,所确定的问题是否可以解决,系统方案在经济上、技术上和操作上是否可以接受。可行性分析主要包括经济可行性、技术可行性、使用可行性和法律可行性^[14]。

经济可行性指估计开发费用以及新系统可能带来的收益,将两者进行权衡,看结果是否可以接受。技术可行性是指对要求的功能、性能以及限制条件进行分析,是否能够做成一个可接受的系统。所考虑的因素通常还应包括开发的风险,是否能够得到需要的软件和硬件以及一个熟练的有能力的开发队伍,与系统开发有关的技术是否足以支持系统的研制。使用可行性是判断系统的使用方式在该用户组织内是否可行。法律可行性是指判断系统是否违反现有的法律。

由于该系统不用于商业故无须进行经济可行性分析。

技术可行性分析:

(1) 因目前大多数 PC 机的操作系统为 Windows,因此,使用 Windows 操作系统为平台。

(2) 由需求功能可知本系统对汽车驱动桥专业知识要求较高。由于汽车工程系有众多的车辆工程专业的教授、专家和研究生,并且与我们合作的四川建安工业有限公司能提供很好的技术支持,能保证驱动桥专业技术的需求;

(3) 使用 MS VC++6.0 的 MFC,可方便界面的制作;ADO(ActiveX Data Object)是 Microsoft 数据库应用程序开发的,使用它可以满足对数据库的编程;

(4) I-DEAS 是美国 SDRC 公司开发的 CAD/CAM 软件,该软件是高度集成化的 CAD/CAE/CAM 软件系统,它有强大的三维建模、有限元分析等功能,并且提供了与高级语言的接口语言,能在高级语言中很好地操纵它,可以满足功能需求中的三维建模和有限元分析的要求。

使用可行性分析:

该系统是运行在 Windows 操作系统上,使用 Windows 标准界面,操作简单;使用此系统还需要一定汽车驱动桥专业知识。由于用户均是汽车驱动桥专业设计人员,因此系统在该用户组织中使用应该是可行的。

2.1.3 系统数据流分析

数据流图(data flow diagram,DFD)描述系统的数据流向,是一种用来表示信息流程和信息变换过程的图解方法,它把系统看成是由数据流联系的各种功能的组合,它可以方便地描述用数据流的流动联系的各种功能,通过每种功能的输入数据/输出结果,进而表示现有系统或待开发系统的功能^[14]。数据流图通常用分层画法。把整个软件系统看成一个整体,可以画出系统的输入输出图,这张图称为顶层图。将顶层图

的加工分解成若干个加工,并用数据流将这些加工连接起来,使得顶层图中的输入数据流经一连串的加工处理后变换成顶层图的输出数据流,这张图称为0层图。

由2.1.1的功能需求分析可知,驱动桥设计系统与I-DEAS系统紧密接合,因此我们在分析数据流图时将驱动桥设计系统与I-DEAS系统看成一体,但为了更清楚地知道驱动桥设计系统的输入/输出数据,在画顶层图时我们将两者分开了。本系统从外界接受的信息主要来自驱动桥设计人员,它包括:选型参数,驱动桥设计原始参数,有限元分析原始数据,评估要求。

另外,在生成所设计的驱动桥零件图时I-DEAS系统需要读取零件库中的参数化零件模型,它以I-DEAS系统所识别的数据库的形式存在,因此,这个零件库也是系统接收的数据源。但是,零件库数据输出直接指向I-DEAS系统,驱动桥设计系统不对它进行读取。

同样可以分析出,系统向外界输出数据主要有:不正确选型方案,有限元分析结果文档,错误数据,评估结果和驱动桥三维模型。而驱动桥设计系统向外界输出的数据主要有:不正确选型方案,建模接口文件,有限元接口文件,评估结果。如图2.1所示。

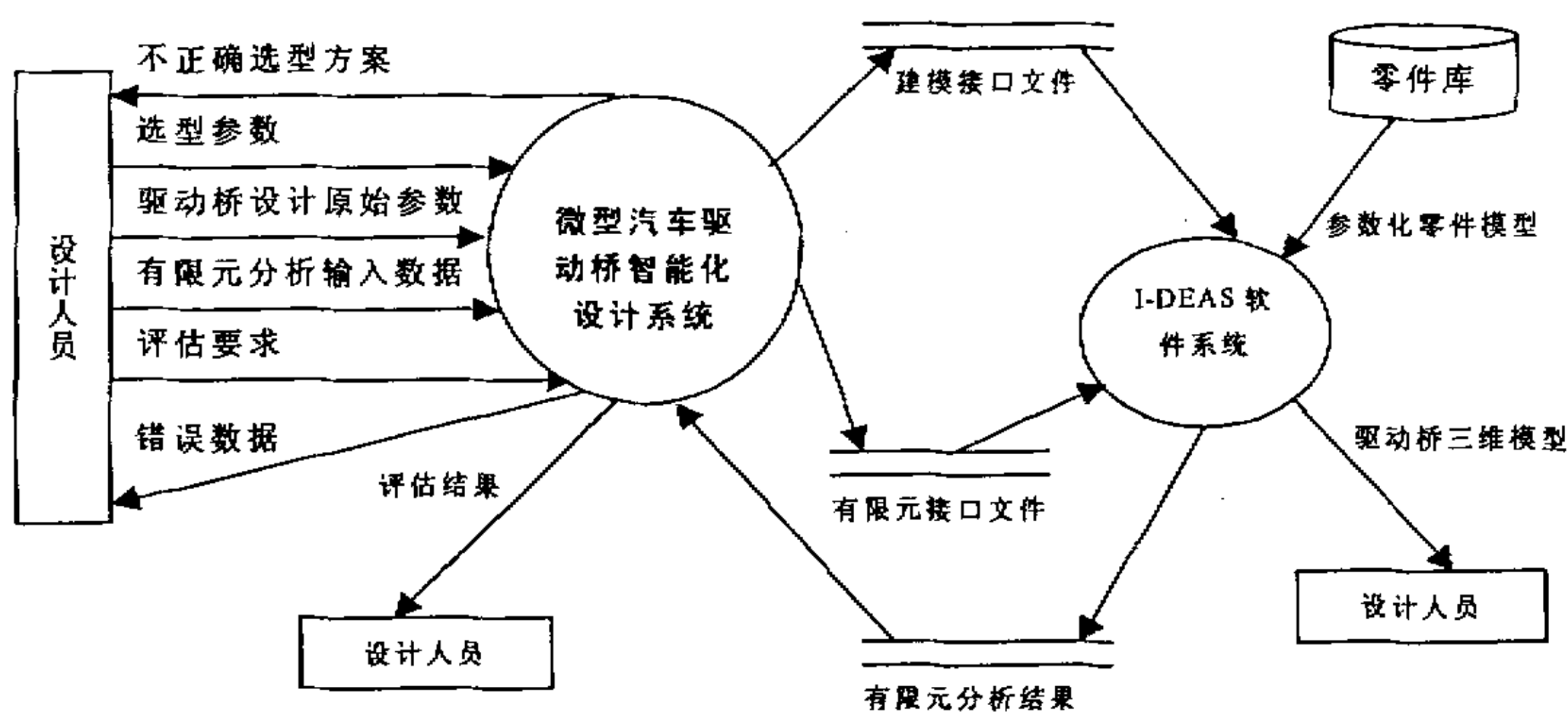


图 2.1 顶层图

不正确选型方案说明设计者输入的选型参数产生的方案不符合逻辑或系统不支持。建模接口文件主要记录与建模有关的参数,如零件几何尺寸值、零件号和零件数。有限元接口文件主要记录与I-DEAS系统进行有限元分析有关的数据。建模接口文件和有限元接口文件将在5.2节系统的接口设计中详细讨论。有限元分析结果文件记录了所设计的零部件的有限元分析结果数据,它可以被系统读取并根据评估要求作出评估从而生成评估结果。

根据功能需求和顶层图细化出 0 层图如图 2.2。

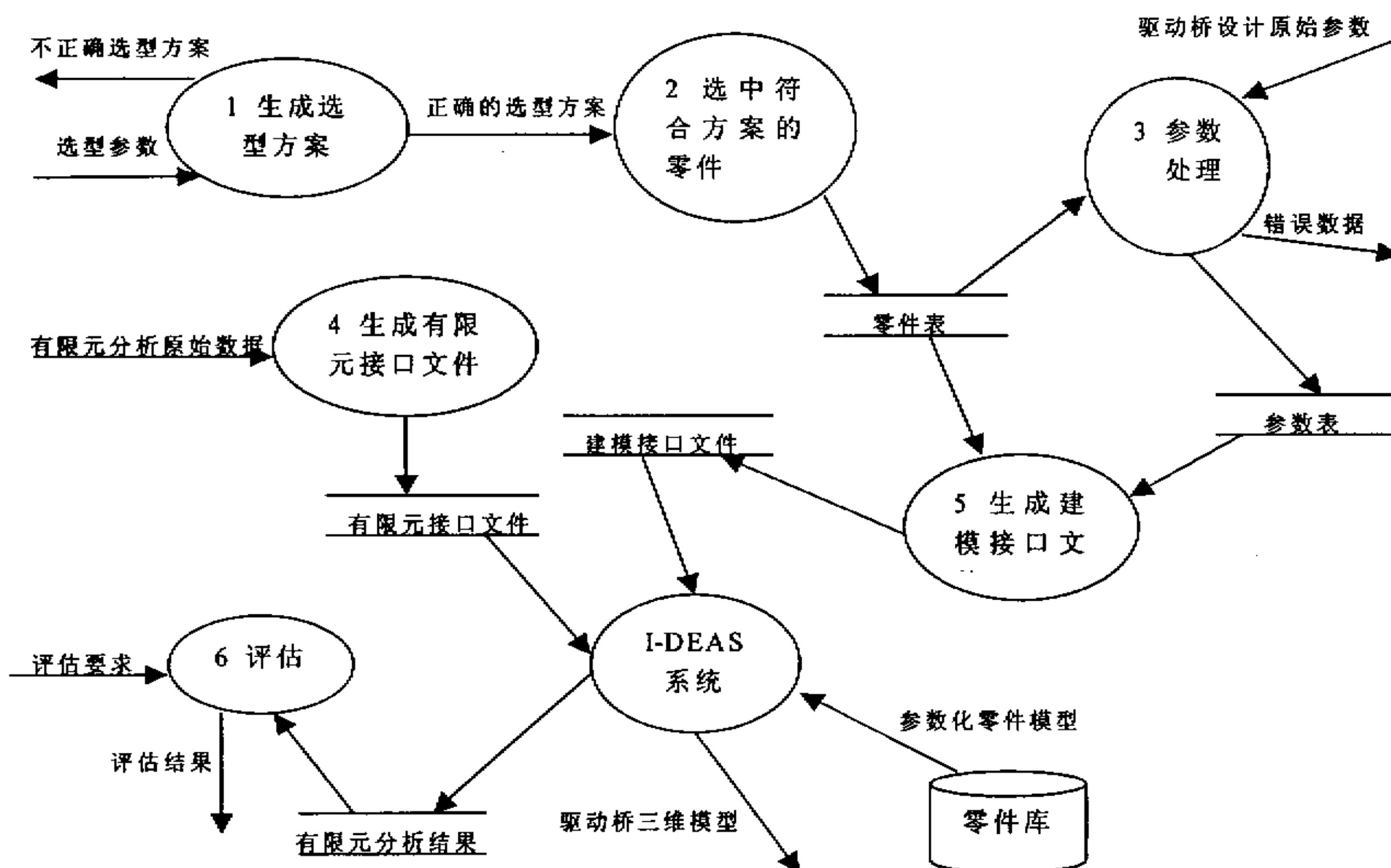


图 2.2 0 层图

由图可知，微型车驱动桥智能化设计系统被分解成六个加工：生成选型方案，选中符合方案的零件，参数处理，生成有限元接口文件，生成建模接口文件，评估。在数据流程图 (DFD) 中用圆圈表示，所谓加工是指对数据执行某种操作或变换，是把输入数据变成输出数据的一种变换。同时，在微型车驱动桥智能化设计系统内部还产生了两个文档：零件表，参数表。

六个加工主要功用阐述如下：

加工 1：处理设计人员输入的选型参数，产生相应的选型方案。由于设计人员的某些选型方案是不符合逻辑的或者此选型方案驱动桥设计系统暂不支持，因此输出的选型方案有两种可能，正确的和不正确的；若是正确的则将选型方案送到加工 2，若不正确，则提示设计者出错信息。为了知道系统支持哪些选型方案，此加工需要一个选型方案表来查询可用的选型方案。有关驱动桥选型方案在软件系统中的实现将在 3.1 节阐述。

加工 2：在输入一个正确的选型方案之后，根据选型方案选中零件库中生成此方案所需的零部件，这里零件库中的零件以零件表的形式体现出来，零件表中每个零件有一个零件库中零件的 ID 号。选中一个零部件就在零部件表中作一选中标志。具体

实现将在 3.1 节和 3.2 节阐述。

加工 3: 是整个系统数据处理的核心部分。设计者将输入所有设计所需要的原始参数,如发动机最大扭矩、车轮滚动半径、最低档速比、传递效率等等。驱动桥所有零部件几何尺寸的计算也在此加工生成,并将其写入到参数表中。它需要参数值智能更新算法来完成参数的计算。算法实现将在 3.3 节阐述。

加工 4: 生成有限元接口文件。I-DEAS 系统是根据此加工处理后生成的有限元接口文件来进行有限元分析的,将在 5.3 节阐述。

加工 5: 生成建模接口文件。I-DEAS 系统是根据此加工处理后生成的建模接口文件来进行建模的,也将在 5.3 节阐述。

加工 6: 根据用户的要求并按照设计完的驱动桥分析结果作出评估。

数据流图(DFD)是进行功能模块划分的依据,2.2 节中我们将按照图 2.2 来进行功能模块划分。

2.2 系统总体设计

总体设计的任务是确定软件系统的结构,进行模块划分,确定每个模块的功能、接口以及模块间的调用关系^[4]。功能模块划分时必须按照以下原则进行,以保证模块分解尽可能理想:

- (1) 满足信息隐蔽原则。
- (2) 尽量使得模块的内聚度高、模块间的耦合度低。
- (3) 模块的大小适中。
- (4) 模块的调用深度不宜过大。
- (5) 模块的扇入应尽量大,扇出不宜过大。一个模块的扇入是指直接调用该模块的上级模块的个数。一个模块的扇出是指该模块直接调用的下级模块的个数。扇入大表示模块的复用程度高,扇出大表示模块的复杂度高。
- (6) 设计单入口和单出口的模块。
- (7) 模块的作用域应在控制域内。
- (8) 模块的功能应是可预测的。

我们将用结构图(structure chart,SC)来描述模块之间的关系。结构图中用矩形来代表模块,并标示模块名,模块到模块间的连线表示调用,连线边上可以标明数据信息。

2.2.1 功能模块划分

按照前节画出的数据流图(DFD)进行变换分析。先进行第一级分解,可以得到图 2.3 所示的功能模块的划分:接收输入信息,参数转换控制,数据输出模块。接收输入信息模块用来接收所有的输入数据,数据转换控制模块用来实现输入到输出的变换,数据输出模块用来产生所有的输出数据。

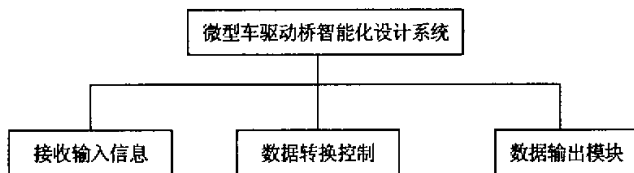


图 2.3 模块总体划分

图 2.4、2.5、2.6 对系统进行了第二级分解。由数据流图 (DFD) 的输入流以及对输入流进行辅助性加工的加工可以得出, 接收输入信息模块可以分解为: 生成选型方案, 接收有限元原始参数, 输入评估要求, 驱动桥设计数据输入。其中, 生成选型方案是对设计者输入的选型参数进行辅助加工生成的, 因此生成选型方案模块又调用选型参数输入模块, 如图 2.4 所示。

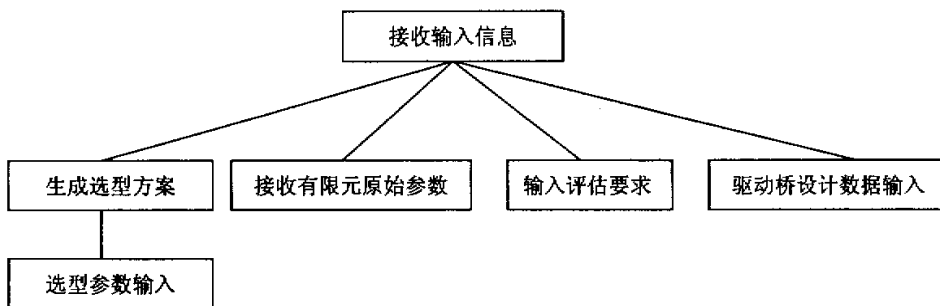


图 2.4 接收输入模块划分

选型参数输入主要功能是为设计者提供一个输入选型参数的界面, 生成选型方案模块主要实现加工 1 所描述的功能。输入评估要求模块、驱动桥设计数据输入模块和接收有限元原始参数模块均为用户提供一个接受数据的界面。

数据转换控制模块可以分解为: 选择符合选型方案的零件, 参数推理计算模块, 生成有限元接口文件, 生成建模接口文件, I-DEAS 系统, 评估模块。其中, I-DEAS 系统主要完成的功能是驱动桥三维建模和零部件有限元分析。如图 2.5 所示。

选择符合选型方案的零件模块主要实现加工 2 功能, 参数推理计算模块主要实现加工 3 功能, 生成有限元接口文件模块主要实现加工 4 功能, 生成建模接口文件模块主要实现加工 5 功能, 评估模块主要实现加工 6 功能。I-DEAS 系统是一个单独的软件系统, 在这里我们仅把它看成是整个系统的一个能完成驱动桥三维建模和零部件有限元分析的功能子模块。

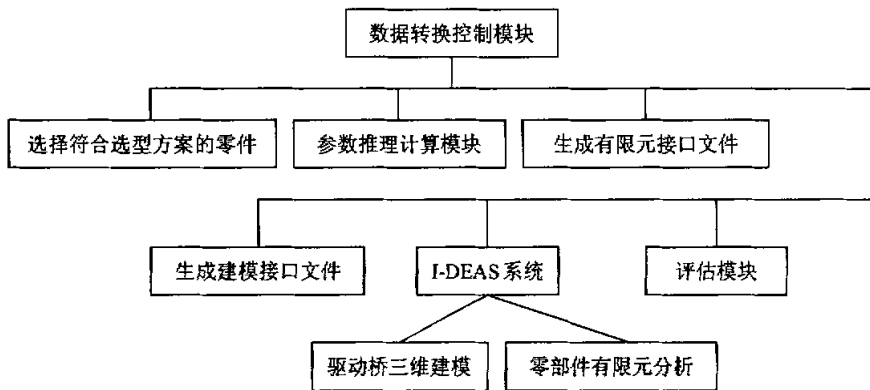


图 2.5 数据转换控制模块划分

数据输出模块可以分解为：选型方案出错，参数处理错误，有限元分析结果，驱动桥三维建模显示，评估结果。如图 2.6 所示。选型方案出错模块主要功能是输出错误信息，提示用户从新输入选型参数。参数处理错误模块主要功能是输出驱动桥设计原始参数错误信息，它可能是输入参数有逻辑错误或校核后不符合错误(如齿轮强度校核不满足、离地间隙不满足等等)。有限元分析结果模块主要是由 I-DEAS 系统按照有限元分析接口文件计算并输出的。驱动桥三维建模显示模块也是由 I-DEAS 系统按照建模接口文件输出的。评估结果模块主要功能是输出评估结果，以供设计者参考分析是否达到期望要求。

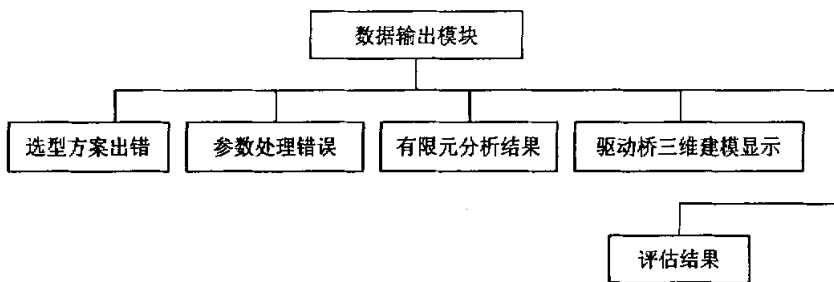


图 2.6 数据输出模块划分

功能模块的划分是详细设计的依据,我们将在第 3 章对主要的功能模块进行详细设计。

2.2.2 系统的软硬件支撑环境

硬件环境为：

- (1) 微机要求主频 166MHz 以上；

- (2) 3D 图形加速卡;
- (3) 硬盘 4G 以上;
- (4) 内存 64M 以上。

软件环境:

- (1) 操作系统: Windows NT 4. 0/ Windows2000;
- (2) 数据库: MS Access 2000;
- (3) 编程语言: VC++6.0;
- (4) 图形处理软件: I-DEAS 集成软件系统。

2.2.3 系统界面

系统软件界面体现了一个软件的设计水平和风格,因此在设计时要满足一定的原则^[5]:

(1) 界面设计的一致性。一致性原则就是软件所有界面的总体风格包括布局、色彩、文字必须一致。

(2) 常用功能操作的便捷性。常用操作的使用频度大,应该减少操作序列的长度。例如,文件的常用操作如打开、存盘、另存等设置快捷键,这样,不仅会提高用户的工作效率,还使得界面在功能实现上简洁和高效。

(3) 较好的容错能力。系统要有错误处理的功能。在出现错误时,系统应该能检测出错误,并且提供简单和容易理解的错误处理的功能。错误出现后系统的状态不发生变化,或者系统要提供错误恢复的指导。

(4) 提供信息反馈。对操作人员的重要操作要有信息反馈。对常用操作和简单操作的反馈可以不作要求,但是对不常用操作和至关重要的操作,系统应该提供信息的反馈。

(5) 操作的可逆性。可逆性操作是指可以返回重新执行的操作,这对于不具备专门知识的操作人员相当有用。可逆的动作可以是单个的操作,或者是一个相对独立的操作序列。

(6) 具有良好的联机帮助功能。对于熟练用户来说,联机帮助并非必不可少,但是对于大多数不熟练用户来说,联机帮助具有非常重要的作用。

界面设计主要包括主界面、数据输入界面和数据输出界面等的设计。主界面包括菜单、工具栏等,菜单给出了系统所能实现的所有功能的入口界面。工具栏是菜单的快捷方式。

本软件系统的主界面框架如图 2.7:

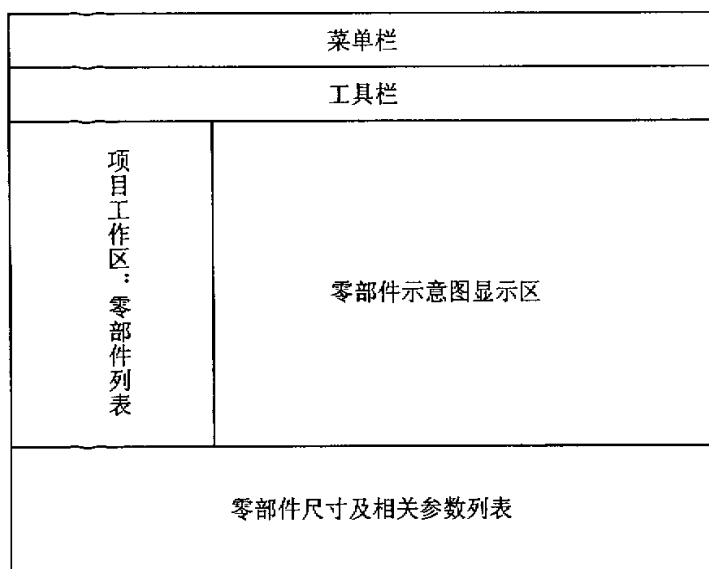


图 2.7 主界面框架

主界面的顶部是菜单栏，菜单栏下方是与菜单栏对应的工具栏；主界面左部是项目工作区，它以树的形式将项目名以及此项目包含的所有零件名列出；右部是图形显示区，它的作用是显示各零部件的三视图以及立体图，使设计者清楚的知道零件每个尺寸对应的值以便设计者对不理想的地方进行加工修改；主界面底部是零件尺寸及相关参数列表，它其实也是一个参数输入界面，当设计者想改动某些参数时，可以直接在列表中输入。

项目工作区、显示区、零件尺寸及相关参数列表之间是相关联的，点击每个零件节点时，零件尺寸及相关参数列表就显示此零件的所有几何尺寸以及与该零件相关的部分参数，同时在显示区显示对应的零件图，显示区根据工具栏中选择的视图显示不同的视图。

菜单设计如图 2.8。

工程	项目	编辑	查看	任务	设置	帮助
新建工程	新建项目	更新相关参数	工具栏	有限元分析	系统初始化	关于系统
打开工程		添加删除零件	状态栏	设计评估	更改参数默认值	联机文档
保存工程		修改零件尺寸	工作区			
工程另存		编辑零件库	尺寸列表			
退出			视图>>>	三视图		
			生成实体	正视图		
			启动 I-DEAS	顶视图		
				侧视图		
				立体图		

图 2.8 菜单设计

3 微型汽车驱动桥智能化设计系统详细设计

在第2章中我们对系统进行模块划分,这一章我们将对各主要模块进行更具体的详细设计。

详细设计的任务,是为软件结构图中的每一个模块确定所采用的算法和块内数据结构,用某种选定的表达工具给出清晰的描述,表达工具可以由设计人员自由选择,但它应该具有描述过程细节的能力,而且便于直接翻译成程序设计语言的源程序。

详细设计的主要描述工具主要有图形描述工具、语言描述工具、表格描述工具。常用的图形描述工具有程序流程图、盒图和问题分析图。典型的语言描述工具是PDL (Program Design Language)^[14]。典型的表格描述工具是判定表和判定树。

3.1 驱动桥选型的解决思路

3.1.1 什么是驱动桥选型

在一般的汽车结构中,驱动桥包括主减速器、差速器、驱动车轮的传动装置及桥壳等部件^[16]。对于各种不同类型和用途的汽车,正确地确定上述机件的结构型式并成功地将它们组合成一个整体——驱动桥,乃是设计者必须首先解决的问题。

本软件系统的驱动桥选型主要解决以下几个问题:

- (1) 选型的主要内容。即哪些机件的结构型式是设计者需要选择的。
- (2) 如何在程序中表示设计者输入的选型参数。即如何构造选型方案数据结构,这其实是对功能模块中的生成选型方案模块的详细设计。
- (3) 如何检查选型方案可用性。

3.1.2 选型的主要内容

考虑到以后系统功能的扩展,设计者首先选择是设计驱动桥还是非驱动桥,为以后加入非驱动桥的设计功能作铺垫。若是选择设计驱动桥则要对驱动桥的每个零部件的结构型式进行选择。驱动桥的结构形式,按其总体布置来说,共有三种,即普通的非断开式驱动桥,带有摇摆式半轴的非断开式驱动桥和断开式驱动桥。对于微型车我们可以简单地分为断开式和非断开式两种。

(1) 对于主减速器的选型主要有:主减速器齿轮类型、主动锥齿轮支承型式、从动锥齿轮支承型式、主减速器减速型式。

① 在现代汽车的驱动桥上,采用的最广泛的主减速器齿轮形式,是“格里森”(Gleason)制或“奥利康”(Oerlikon)制的螺旋锥齿轮和双曲面齿轮。

② 现代汽车主减速器主从动锥齿轮的支撑形式有以下两种:悬臂式,骑马式。

③ 主减速器减速型式可分为:单级减速、双级减速、双速减速、单级贯通、双级贯通等。

(2) 对于差速器的选型主要是对差速器结构形式和行星齿轮数目的选择。差速器结构形式主要有对称式圆锥行星齿轮差速器和防滑差速器。对于微型汽车驱动桥我们主要考虑对称式圆锥行星齿轮差速器。行星齿轮数目有三种：1 个行星齿轮、3 个行星齿轮、4 个行星齿轮。

(3) 驱动车轮的传动装置中主要对半轴的类型进行选择。根据驱动桥外端轴承装置方法的不同，半轴分为：半浮式，3/4 浮式和全浮式。

(4) 桥壳结构型式大致可分为三种：可分式，整体式和组合式。其中，可分式桥壳按其组合的件数又可分为二段可分式和三段可分式两种。

(5) 轴承的选择。主要有：主动齿轮远端轴承、主动齿轮近端轴承、半轴轴承、差速器轴承。对于微型车驱动桥，轴承主要有两种：圆锥滚子轴承和深沟球轴承。

以上所述大致可以包括设计者选型的内容，但不能排除还有其它的选型内容，因此必须考虑到未来若发现其它的选型内容，能够在程序中容易加进去，这将在设计选型方案数据结构时考虑到。

3.1.3 构造选型方案数据结构

前面我们找出了选型内容，但这只是文字表达，我们要把它们变成程序中可识别的方式。我们用一个结构体表示选型方案如下：

```
//选型方案的结构体
struct designtype
{
    int isdrive;           //是否驱动桥
    int drivetype;         //驱动桥类型
    int geartype;          //主减速器齿轮类型
    int gearbearing;       //主动锥齿轮支承型式
    int gearbearing2;      //从动锥齿轮支承型式
    int detype;            //主减速器减速型式
    int differencetype;    //差速器结构型式
    int planetnum;         //行星齿轮数目
    int shafttype;         //半轴型式
    int shelltype;         //桥壳结构型式
    int farbearing;        //主动齿轮远端轴承
    int nearbearing;       //主动齿轮远端轴承
    int differbearing;     //差速器轴承
    int shaftbearing;      //半轴轴承
};
```

designtype 的某个数据成员在选择不同的结构型式时它的值是不同的。例如，选

择设计驱动桥时 isdrive=1,选择非驱动桥时 isdrive=2。

在设计者给出了选型方案之后,我们需要从预先建立起来的零件库中选取符合此选型方案的零件,以作为将来输出驱动桥三维图的参数化模型。所谓参数化模型是指此模型的结构型式是确定的但尺寸值是可变的,即输入不同的尺寸值时,模型会随之而变。所以这里的零件库所存放的是零件的参数化模型。

显然,零件库中的每一个零件有可能只满足一种选型方案,也有可能满足多种选型方案。

例:某一主减速器齿轮模型可能满足选型方案:

{是驱动桥,断开式,双曲面齿轮,悬臂式,悬臂式,单级减速器,对称式圆锥行星齿轮差速器,2,半浮式,三段可分式,圆锥滚子轴承,圆锥滚子轴承,圆锥滚子轴承,深沟球轴承} 记为 {1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1} (其中的每一项依次对应 designtype 中的每一项);

当然也可以满足选型方案:

{是驱动桥,断开式,双曲面齿轮,悬臂式,悬臂式,单级减速器,对称式圆锥行星齿轮差速器,2,半浮式,整体式,圆锥滚子轴承,圆锥滚子轴承,圆锥滚子轴承,深沟球轴承} 记为 {1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1}。

两种方案中不同的是桥壳的结构型式:第一种是三段可分式,第二种是整体式。我们可以给定零件库中的每个零件所满足的所有方案。例如,上面的主减速器齿轮模型可以满足的方案为:

{ {1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1},
{1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1} }。

如果某个零件可以满足十几种方案的话我们就需要列出十几组数据,这样记录数据很不方便而且查找也不方便。如何解决呢?最好能只用一组数据就能表达出来。

由前面可知,驱动桥类型有两种:断开式和非断开式;用二进制表示:01, 10。主减速器减速型式有五种:单级减速,双级减速,双速减速,单级贯通,双级贯通;用二进制表示:00001, 00010, 00100, 01000, 10000。同理可得出以下表示法(下标 2 表示二进制, 10 表示十进制):

是否驱动桥: (01, 10)₂ 即 (1, 2)₁₀ ;
驱动桥类型: (01, 10)₂ 即 (1, 2)₁₀ ;
主减速器齿轮类型: (01, 10)₂ 即 (1, 2)₁₀ ;
主动锥齿轮支承型式: (01, 10)₂ 即 (1, 2)₁₀ ;
从动锥齿轮支承型式: (01, 10)₂ 即 (1, 2)₁₀ ;
主减速器减速型式: (00001, 00010, 00100, 01000, 10000)₂

即 $(1, 2, 4, 8, 16)_{10}$;

差速器结构型式: $(01, 10)_2$ 即 $(1, 2)_{10}$;

行星齿轮数目: $(001, 010, 100)_2$ 即 $(1, 2, 4)_{10}$;

半轴型式: $(001, 010, 100)_2$ 即 $(1, 2, 4)_{10}$;

桥壳结构型式: $(0001, 0010, 0100, 1000)_2$ 即 $(1, 2, 4, 8)_{10}$;

主动齿轮远端轴承: $(01, 10)_2$ 即 $(1, 2)_{10}$;

主动齿轮远端轴承: $(01, 10)_2$ 即 $(1, 2)_{10}$;

差速器轴承: $(01, 10)_2$ 即 $(1, 2)_{10}$;

半轴轴承: $(01, 10)_2$ 即 $(1, 2)_{10}$;

于是我们就可以这样来表示某模型既满足单级减速又满足双级减速:

$00001(\text{单级减速}) + 00010(\text{双级减速}) = 00011(\text{满足单级减速又满足双级减速})$

即 00011 表示既满足单级减速又满足双级减速。其他可以同样表示: 即只要把所有的表示满足的结构型式的二进制数相加即可。

现在我们可以将例子中的主减速器齿轮模型满足的方案:

$\{1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1\}$,

$\{1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1\}$

表示为:

$\{01, 01, 01, 01, 01, 01, 01, 01, 01, 11, 01, 01, 01, 01\}$

我们将零件库中的某个零件模型满足的所有方案按照以上方法表示成一数组, 称这一数组为此零件模型的满足方案, 记为 PartDesignType。

现在的问题是如何判断零件库中的某个零件模型的满足方案是否符合设计者给出的选型方案。

记设计者给出的选型方案为 UserDesignType, 那么可以证明当下式成立时 PartDesignType 符合 UserDesignType:

$$UserDesignType \& PartDesignType = UserDesignType \quad (3-1)$$

证明: 所谓 PartDesignType 符合 UserDesignType, 是指零件库中的某个零件模型满足的所有方案 PartDesignType 中包括了设计者给出的选型方案 UserDesignType。

记 UserDesignType 为 U, PartDesignType 为 P; U, P 用二维数组表示为:

$U = u[n][m], P = p[n][m]$

其中: n 为 struct designtype 的成员个数;

$m = \max\{\text{designtype 的第 } i \text{ 个成员可取值个数} | i = 1, 2, \dots, n\}$;

$u[i] \in \text{正整数}, p[i] \in \text{正整数}, i = 1, 2, \dots, n$;

$u[i][j] \in \{0, 1\}, p[i][j] \in \{0, 1\}, i = 1, 2, \dots, n, j = 1, 2, \dots, m$;

即, $u[i]$ 表示 U 的第 i 个数据成员的值, $u[i][j]$ 表示 $u[i]$ 表示成二进制数时从左到

右的第 j 个二进制位值； $p[n][m]$ 数组意义同理。则：

$$U \& P = U \Leftrightarrow u[i] \& p[i] = u[i] \quad (i=1, 2, \dots, n)$$

又由位运算法则

$$\Rightarrow u[i] \& p[i] = u[i] \Leftrightarrow u[i][j] \& p[i][j] = u[i][j] \quad (i=1, 2, \dots, n; j=1, 2, \dots, m)$$

由 U 的定义可令：

$$u[i][j] = \begin{cases} 1 & j=s \\ 0 & j \neq s \end{cases} \quad (i=1, 2, \dots, n; j=1, 2, \dots, m; 0 < s \leq m \text{ 且 } s \text{ 为自然数})$$

则由位运算法则得：

$$u[i][j] \& p[i][j] = \begin{cases} p[i][j] & j=s \\ 0 & j \neq s \end{cases}$$

$$\therefore u[i][j] \& p[i][j] = u[i][j] \Leftrightarrow p[i][j] = 1 \quad (j=s) \quad \text{即} \quad p[i][s] = 1$$

综上有：

$$U \& P = U \Leftrightarrow p[i][s] = 1 \quad (i=1, 2, \dots, n; 0 < s \leq m \text{ 且 } s \text{ 为自然数})$$

而 $p[i][s] = 1$ 表示 $p[i]$ 表示成二进制数时从左到右的第 s 个二进制位值为 1；

由 P 的定义可知 P 符合 U ；

证毕。

对于前面的例子，假设设计者给出的选型方案为：

$$a = \{01, 01, 01, 01, 01, 01, 01, 01, 01, 01, 01, 01, 01, 01\}$$

主减速器齿轮模型满足方案为：

$$b = \{01, 01, 01, 01, 01, 01, 01, 01, 01, 11, 01, 01, 01, 01\}$$

则用(3-1)判断如下：

$$a \& b = \{01, 01, 01, 01, 01, 01, 01, 01, 01, 01, 01, 01, 01, 01\} = a$$

故，此零件满足选型方案 a 。

3.1.4 如何检查选型方案可用性

在程序中用 `bool getFuctionType()` 函数来实现。函数框架如下：

```
bool getFuctionType()
{
```

```

struct designtype type[]=
{{1,1,1,1,1,1,1,1,1,1,1,1},
{1,1,2,1,1,1,1,1,1,1,1,1},
{1,1,1,1,1,1,1,1,4,1,1,1},          //包含本软件可以实现的方案
{1,1,1,1,1,1,1,1,8,1,1,1}, .....}; //这里假设有 10 种可用方案

for(int j=0;j<10;j++)
{
    //判断设计者给出的方案 usertype
    if(usertype 与 type[j]相同) //是否与 type[j]符合
    {
        fuctiontype=j+1;        //记录第几种可用方案
        return true;            //表示有可用方案返回 true
    }
}
return false;                  //表示无可用方案返回 false
}

```

我们将“3.3 尺寸约束函数库的建立方法”中讨论全局变量 fuctintype 的作用。

3.2 数据库设计

3.2.1 数据库设计方法概述

由第 2 章概要设计我们可以看出系统在运行时有很多的数据需要存储和读取,如参数表、零件表等。特别是在参数计算推理模块中要频繁地进行数据查询、读取、写入,若只是用一般的文件记录数据操作起来非常不方便,因此需要利用数据库技术。

数据库表的设计是很重要的,设计的好坏直接影响到数据操作的效率和正确性。数据库的设计方法很多,我们用统一建模语言(Unified Modeling Language ,UML)进行设计^[17]。UML 是一种通用的模拟语言,它可以用于确定、展示和记录软件系统,统一建模语言中的图形标记,尤其适用于面向对象的软件设计。将它用于数据库设计主要是将设计好的持久对象转化成数据库模型。所谓持久对象是指,需要转化成一定格式的数据存放到磁盘或其他媒介的对象,以便将来提取使用^[17]。

数据库一般可分为四种模型^[17]: 关系数据库(relational database)、扩展关系数据库(extended relational database)、函数数据库(functional database)和面向对象数据库(object-oriented database)。本系统的数据库是关系数据库,对于关系数据库需要将持久对象映射成表。以下我们将先设计出持久对象,然后将它们映射表。

3.2.2 整体对象设计

对系统功能需求进行分析可知,数据库中应主要记录零部件信息和各类参数信息。各类参数主要包括:零件尺寸参数,尺寸变量参数,原始输入参数。尺寸变量参数是指可变的经验几何参数,它不是几何图形中必须的尺寸参数但它能导致零件尺寸参数的改变。它的含义是这样的:软件系统在自动推导零件几何参数时,会发现零件几何参数在某个范围内变化都是可行的,则这个变化范围就是尺寸变量参数。尺寸变

量参数的作用将在 3.3 更新算法实现中具体讨论。我们可以画出下面的整体对象设计图：

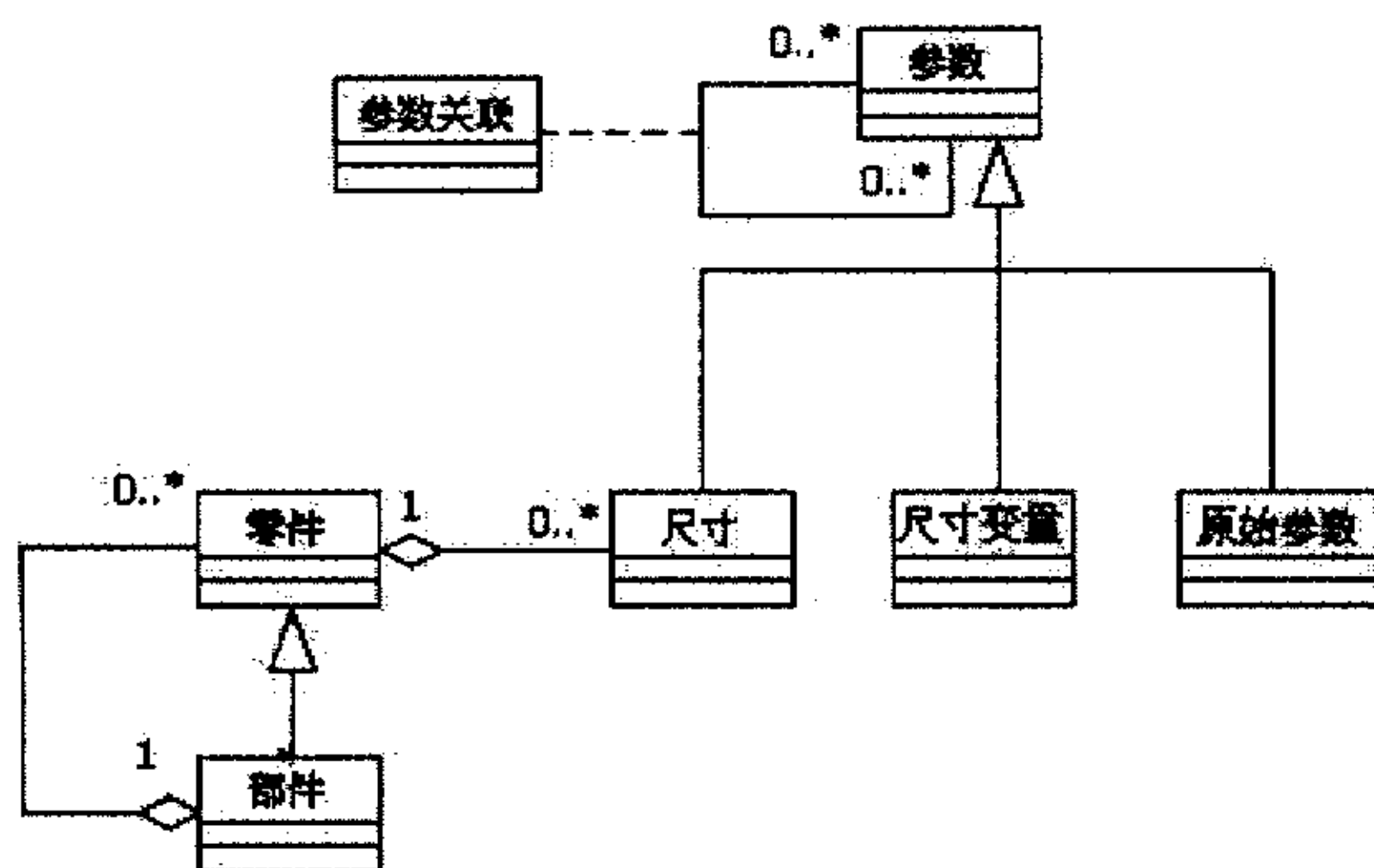


图 3.1 整体对象设计

上图中，每一框图代表一个类，单向箭头代表类的继承关系，双向箭头表示集团组成关系，“1”表示唯一的一个，“0..*”表示零或多个。图中部件类从零件类继承并且每个部件由多个零件组成，每个零件包括多个尺寸；尺寸类、尺寸变量类和原始参数类从参数类继承；参数类与自身有关联，比如某个参数与多个参数之间存在函数关系，或则多个参数按照某种计算法则可以推出另一个参数，所以上图中标明“0..*”（零或多个）参数对应“0..*”。参数关联类负责处理参数类之间的连接关系。

3.2.3 详细对象设计

在整体对象设计完成之后，就要对每一个类进行详细设计。详细设计主要目的是设计出每个类的属性以及对属性进行的运算。每个类的详细设计结果用三个格子的矩形框表示，最上层一格写入类名，第二格写入此类所有的属性，第三格是类的成员函数即对属性进行的运算。我们之所以要把每个类详细的写出来，是因为每个类的属性将直接被映射到关系数据库表的字段(field)中，它是设计数据库表的依据。

按照图 3.1 的整体对象设计结果，可以知道每个类应该具有的属性。例如，参数类，一个参数一般应具有参数名、参数值，又因为图中参数自身有关联，所以再加一个参数关联对象属性。类的成员函数主要有构造函数(constructor)、提取和询问运算(get & query operations)、编辑修改(editing)。因此，可以详细设计每个类如图 3.2——3.7，图中给出了各个类每个属性的含义，并给出了部分成员函数的功能描述。图中的向下关联对象是指一个包含所有服务于该参数的参数的关联对象；向上关联对象是指一个包含所有该参数所服务的参数的关联对象。

CParameter //参数类名
m_name: String //参数名 m_value: double //参数值 m_relation: Container //该参数的关联对象
<pre> 《constructor》 CParameter(); 《get & query operations》//提取和询问运算 getName(): String //提取参数名 getValue(): double //提取参数值 getRelation(): CObArray //提取关联 《editing》 //编辑修改 setName(name: String) //设置参数名 setValue(value: double) //设置参数值 setRelation(relation: CParaRelation) //设置关联对象 </pre>

图 3.2 参数类

CPart //零件类
m_name: String //零件名 m_dimension: Container //零件的尺寸集 m_parts: CParts //零件所属的部件 m_num: int //尺寸数 m_style: int //零件所属的结构类型,表示成二进制格式每一位表示某一类型
<pre> 《constructor》 CPart(): CPart 《get & query operation》 getName(): String getDimension(index: int): CDimension //提取第 index 个尺寸对象 getParts(): CParts getNum(): int 《editing》 setName(name: String) addDimension(dimension: CDimension) removeDimension(dimension: CDimension) setParts(parts: CParts) setNum(num: int) </pre>

图 3.3 零件类

CDimension //尺寸类
m_MotherPart: CPart //尺寸所属的零件
<pre> 《constructor》 CDimension() 《get & query operations》 getMotherPart(): CPart //提取尺寸所属的零件对象 《editing》 setMotherPart(motherpart: CPart) //设置尺寸所属的零件的对象 </pre>

图 3.4 尺寸类

CDimensionVar //尺寸变量类
m_MotherPart: CPart //尺寸所属的零件
<pre> 《constructor》 CDimension() 《get & query operations》 getMotherPart(): CPart //提取尺寸所属的零件对象 《editing》 setMotherPart(motherpart: CPart) //设置尺寸所属的零件的对象 </pre>

图 3.5 尺寸变量类

CParaRelation //参数关联类
m_para: CParameter // 主参数 m_relationtype: bool; //关联方式, 向上关联还是向下关联 m_relationpara: CParameter //关联参数
《constructor》 CParaRelation(): CParaRelation 《get & query operation》 getPara(): CParameter getRelationPara(): CParameter getRelationType(): bool 《editing》 setPara(Para: CParameter)

图 3.6 参数关联类

CParts //部件类
m_part: CPart[*] //部件包含的零件 m_num: int //零件数
《constructor》 CParts(): CParts 《get & query operation》 getPart(index: int): CPart //提取第 index 个零件 getNum(): int 《editing》 addPart(part: CPart) removePart(part: CPart) setNum(num: int)

图 3.7 部件类

3.2.4 对象映射成表格

利用映射法(mapping)可以将类映射到关系数据库。普遍来说, 一个类映射为一个表格, 表的列或字段(field)对应类的属性, 而行或记录(record)对应类的实例(instance)。在表格中一个或几个字段构成主键码(primary key), 其值则是记录的独特标志。从概念上看, 表格就像持久对象的域(extent)。例如, 对于参数类的属性有: 参数名, 参数值, 该参数的关联对象。因此, 参数名、参数值和该参数的关联对象将映射为表的三个字段, 而每一个实际参数为表的记录。但是, 其中的“参数的关联对象”属性是以对象的类型表示的, 而关系数据库中无法表示类对象, 因此需要用另一个表格来模拟这个对象对应的类。映射法则可以总结如下^[17]:

- (1) 带有简单属性的类 A 映射为表格 A, 其主键码是 A_ID。
- (2) 类 A 和 B 之间的二元关联, 并带关联类 L 和多重性 mA 和 mB :
 - 如果 mA>1 和 mB>1, 则连接表格 L 的主键码为(A_ID, B_ID);
 - 如果 mA=1 和 mB>1, 则连接表格 L 的主键码为 B_ID;
 - 如果 mA>1 和 mB=1, 则连接表格 L 的主键码为 A_ID;
 - 如果 mA=1 和 mB=1, 则连接表格 L 的主键码为 B_ID 或 A_ID。
- (3) 聚合、复合和服务类: 可以照二元关联一样映射。
- (4) N 个类之间的 N 元关联并带关联类 L : 映射为 N 个表格, 其各自的主键码

是 $ID_n(0 < n < N+1 \text{ 的整数})$ ；关联类 L 映射为连接表格 L ，主键码为 $(ID_1, ID_2, \dots, ID_n)$ ；如果类的多重性为 1，对应的外键码可以去除。

(5) 继承树：继承树中的每一个类映射为一个表格。所有的表格共用同一主键码。子类从母类继承过来的属性，放在母类的字段中，又下一层子类表格的表格，要加上一个种类 (type) 字段，用来说明每一行属于哪类。

按照上面的映射法则，本驱动桥智能化设计系统数据库共有四个表：参数关联表，参数表，零部件表和满足方案表。各表字段如下。

(1) 参数关联表字段如下：

源参数 ID	目的参数 ID
--------	---------

上表中，(源参数 ID, 目的参数 ID) 构成了参数关联表的主键。源参数 ID 和目的参数 ID 都是外键，它们来自参数表的主键参数 ID。目的参数是由源参数约束的。例如，某一零件的尺寸 a 在设计过程中是由尺寸 b 和尺寸 c 推出的 (b 和 c 有可能是属于此零件也有可能属于其它零件) 并且存在如下函数关系： $a=b+c$ 。那么这里的尺寸 a 就是目的参数，而尺寸 b 和 c 是源参数。在参数关联表中将加入如下两条记录： b, a ； c, a 。而函数 $a=b+c$ 将被加入函数库以便在计算时调用，函数库的建立方法我们将在下一节详细讨论。

由此可见，参数关联表将所有有关联的 (注意不仅仅是函数关联) 参数联系起来了。这样一旦改变了某些参数，就可以通过它搜索出所有与此相关的参数，并对相关的参数作相应的修改，在“3.3 参数值智能更新算法的实现”中我们会看到它是如何工作的。

(2) 参数表字段如下：

参数 ID	参数名	参数值	所属零部件 ID	参数类型
-------	-----	-----	----------	------

上表中，参数 ID 是表的主键。参数类型表示参数的性质，如尺寸参数、尺寸变量参数、原始参数等。所属零部件 ID 是零部件表的主键，表示这个参数属于这个零件，它只是尺寸参数和尺寸变量参数的字段。

参数表记录了每个参数的具体信息，它是计算时的数据来源。例如，上面 $a=b+c$ 中 b 和 c 的值是通过查询参数表得到的。同样，在“3.5 参数值更新算法的实现”中我们会看到它是如何工作的。

(3) 零部件表字段如下：

零部件 ID	类型	零部件名	所属部件 ID	零件数	满足方案 ID	尺寸数	零件库 ID	选中标志
--------	----	------	---------	-----	---------	-----	--------	------

上表中，零部件 ID 是主键。类型是用来区分零件和部件的，因此它只可能有两种值：零件(part)，部件(parts)。所属部件号是零件类的属性，因此只有当类型为零件时才起作用。零件数是指一个部件含有的零件数，是部件类的属性，因此只有当类型为部件时才起作用。

我们在 3.1.3 中已经阐述过满足方案的概念，这里的满足方案 ID 就给出了每个零部件的满足方案。但是满足方案有好几个字段这里表示起来不方便，所以我们单独给出了满足方案表，满足方案 ID 是它的主键。零件库 ID 是零件库中表示零件的唯一编号，在第 5 章中我们将讨论它。

选中标志字段是实现“选择符合选型方案的零件”功能模块所必需的，被选中的零部件的选中标志将被置真值。

(4) 满足方案表字段如下：

满足方案 ID	isdrive	drivetype	...	differbearing	shaftbearing
---------	---------	-----------	-----	---------------	--------------

上表中，满足方案 ID 是主键。它的每个字段依次是 struct designtype 的成员变量。它记录各种零部件的满足方案，并被零部件表调用。

3.3 尺寸约束函数库的建立方法

参数关联表描述的只是某个参数与哪些参数有关系，并没有描述它们之间的运算关系。那么它们之间的运算关系是怎么给出的呢？我们把所有的参数划分为两类：一类是尺寸参数；另一类是所有的非尺寸参数。对于第二类参数的运算处理我们将在 3.4 节专门讨论，这一节我们主要讨论第一类参数运算的实现。

我们知道，在设计零部件时，当需要计算的工作都完成之后就可以开始画零部件的工程图了，我们将尺寸中存在的所有函数关系放入函数库中；但是，在设计中每个尺寸并不只是一个确定的值，它可以在一个允许的范围内变动，这个可变动值就是尺寸变量参数。这样，设计者可以在初步设计之后通过改动尺寸变量参数来调整尺寸值。

当所有的尺寸函数都加入函数库时，就完成了函数库的建立。由于所有的尺寸值都受这个函数库里的函数约束，因此我们称它为尺寸约束函数库。这个函数库建立的正确与否直接关系到设计结果的正确性，因此必须与有经验的驱动桥设计人员共同来完成这个函数库的建立。

考虑到函数库不可能一成不变，而且也不可能绝对没有错误，我们用一个动态链接库单独实现，以便于函数库的修改维护。在主程序中我们用一个入口函数来调用，函数结构如下：

```

//////////<<<<<函数库入口函数>>>>>//////////
//此函数能计算出具体的值，是由公式约束的，调用 DLL 计算参数值函数
double CalculateParaValue(int paraID)
{
    { //设置动态链接库环境 }

    switch (paraID) //paraID 为参数表参数 ID 字段对应的值
    {
    case 101:
        val=(*pf)(currpro.fuctiontype);
        return val;
        break;
        .
        .
        .
    case 3707:
        val=(*pf)(currpro.fuctiontype);
        return val;
        break;
    default:
        return NULLVAL;//if no the paraID then return this value;
    }
}

```

当要计算某个参数的值时，必须得先知道这个参数在参数表中的 paraID, 然后调用入口函数 CalculateParaValue(paraID), 则返回计算结果。函数内部用 switch...case 语句根据 paraID 的值调用动态链接库中相应的函数。动态链接库中每个参数计算函数形式如下：

```

double f212(int functiontype)
{
    double val=NULLVAR;
    switch(functiontype)
    {
    case 1:
        val=Val(3402)+Val(226);
        break;
    case 2:
        val=Val(3402)-2*Val(220);
        break;
        .
        .
        .
    case 10:
        break;
    }
    return val; }

```

函数名的形式为：f 参数 ID(int fuctiontype)。用参数 ID 命名以便入口函数统一调用。我们在 3.1.1 中提到的全局变量 fuctiontype，在这里实现了它的价值。正如前面所述，一个零件可以用在多种选型方案中，但是在不同的方案中它的某些尺寸的计算函数可能不同，因此每一种选型方案需要一套不同的计算函数。

从上面的函数框架可以知道：它计算“参数 ID”为 212 的参数值，它与 ID 为 3402、226、220 的参数有关联。因此参数关联表中一定有这样三条记录：(3402, 212), (226, 212), (220, 212)。若 fuctiontype=1, 则函数将返回 val=Val(3402)+Val(226)的计算结果，函数 Val(int paraID)用来从数据库的参数表中读取参数 ID 为 paraID 的参数值。

3.4 知识约束库的建立

前面我们已经提到，参数划分为两类：一类是尺寸参数；另一类是所有的非尺寸参数。非尺寸参数又可称为知识约束型参数，在产品参数化设计中，这类参数的设计是利用相关领域的专业知识确定的，建立知识约束库的目的是使设计知识能够被表达，从而能够约束设计参数，驱动设计参数的改变。

3.4.1 知识约束的分类

知识约束有两类：关键参数设计计算约束，强度刚度约束。

(1) 关键参数设计计算约束

驱动桥零部件的关键参数是由驱动桥的性能参数设计计算而来的，这种设计计算是依赖于设计师的领域设计知识的^[18]。为此，建立由领域设计知识表达的关键参数设计计算约束，一旦驱动桥的性能参数变化，则零部件的关键参数在此类约束的作用下发生变化。该类约束可以由计算机语言所构造的数据处理过程或函数来表达，如图 3.8 所示。

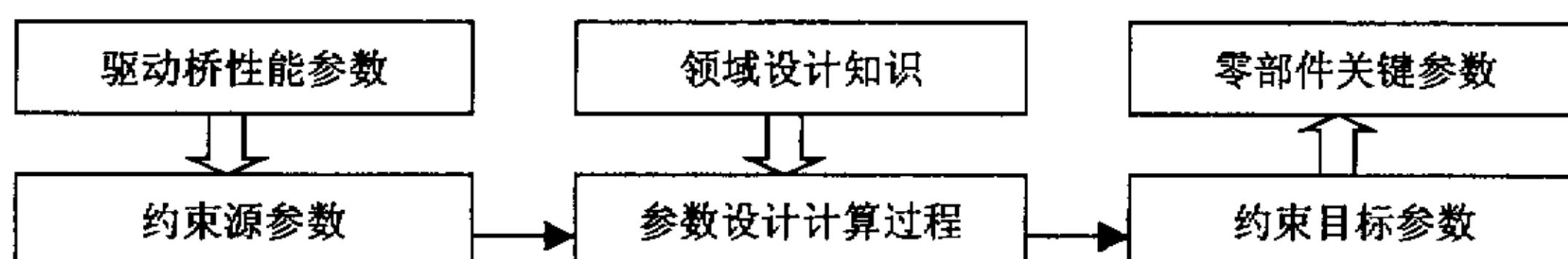


图 3.8 关键参数设计计算约束

(2) 强度刚度约束

在驱动桥结构设计中有许多强度和刚度要求，如 $\sigma < [\sigma]$, $\tau < [\tau]$, $\theta < [\theta]$ 等，相应的参数的设计也牵涉到专业领域内强度和刚度的经验知识。强度和刚度约束是指约束目标参数必须大于或小于约束源参数所构成的表达式值^[18]，即

约束目标参数 $>f$ (约束源参数集)

当约束源参数集中的任一参数被修改时,系统检查该条件不等式是否成立,如果成立,则此次修改是有效的;如果不成立,将警告用户强度刚度条件被破坏,同时系统根据新的约束源参数和表达式 f (约束源参数集)计算出新的约束目标参数,驱动产品模型的修改。

3.4.2 知识约束库的建立方法

知识约束库,其实是大量的函数的集合。每一个函数通过处理专业经验知识、专业计算公式等专业性知识,返回知识约束型参数的比较合理的值。由知识约束的分类可知,知识约束型参数包括:驱动桥性能参数即由设计者直接输入的原始参数,由专业设计知识计算出的中间参数,零部件关键参数。其中,原始输入参数中有一部分,可以是用户直接输入的,也可以通过对驱动桥的描述性要求推理得出一个比较合理的值。例如,轮胎与路面间的附着系数(φ),对于安装一般轮胎的公路用汽车取 0.85,对越野汽车取 1.0,对于安装专门的防滑宽轮胎的高级小轿车可取 1.25。那么我们可以用如下函数推出 φ 的合理取值:

```
double qp(void)
{
    double qp; //qp 表示轮胎与路面间的附着系数  $\varphi$ 
    if(越野汽车)
        qp=1;
    else if(安装防滑轮胎)
        qp=1.25;
    else if(安装一般轮胎)
        qp=0.85;
    else
        qp=0.85;//其他情况
    return qp;
}
```

因此,对于这类参数,驱动桥设计系统需要有一个驱动桥的描述性要求获取界面,我们经归纳总结这些描述性要求主要有:汽车用途,路面条件,轮胎特点,选型方案等。

综上所述,调用知识约束库中的函数主要达到三个目的:第一,按照驱动桥的描述性要求推理出部分参数的合理值;第二,由原始输入参数计算出中间参数;第三,由原始输入参数和计算出的中间参数计算零部件关键参数。

建立知识约束库必须深入了解驱动桥设计计算步骤,从而提取设计知识,并通过分析每个参数是如何得到的,来确定参数的类型。对于驱动桥设计计算的主要步骤如下:

(1) 主减速器设计计算

- ①主减速器齿轮计算载荷;
- ②主减速器齿轮基本参数的选择;
- ③主减速器螺旋锥齿轮与双曲面齿轮强度计算;
- ④主减速器圆弧齿螺旋锥齿轮的几何尺寸计算;
- ⑤主减速器圆弧齿双曲面齿轮的几何尺寸计算;
- (2) 差速器设计计算
 - ①差速器齿轮的基本参数的选择;
 - ②差速器齿轮强度计算;
- (3) 半轴的设计计算
 - ①半浮式半轴的计算: 分纵向力最大时, 侧向力最大时 (侧滑时), 垂直力最大时三种情况;
 - ②全浮式半轴的计算;
 - ③3/4 浮式半轴的计算;
- (4) 桥壳设计计算
 - ①桥壳的静弯曲应力计算;
 - ②在不平路面冲击载荷作用下桥壳的强度计算;
 - ③汽车以最大牵引力行驶时桥壳的强度计算;
 - ④汽车紧急制动时桥壳的强度计算;
 - ⑤汽车受最大侧向力时桥壳的强度计算。

3.4.3 主减速器设计知识的提取过程

下面我们以主减速器设计为例, 详述知识的提取以及参数类型确定的过程。下面参数说明中: “input” 表示为用户输入参数, “reason” 表示此参数可通过对驱动桥的描述性要求推理得出, “check” 表示此参数可以通过查表得到也可以将它看作可推理参数, “fore” 表示此参数类型在它之前的步骤中已给出。

3.4.3.1 齿轮计算载荷

- (1) 按发动机最大扭矩和最低档传动比确定从动锥齿轮计算转矩

$$T_{Ge} = \frac{M_{e\max} \cdot i_1 \cdot i_0 \cdot K_0 \cdot \eta}{n}$$

式中 T_{Ge} ——计算转矩 $N \cdot m$;

$M_{e\max}$ ——发动机最大扭矩 $N \cdot m$, *input*;

i_1 ——变速器一档传动比, *input*;

i_0 ——主减速器传动比, *input*;

K_0 ——超载系数, 对一般载重汽车、矿用汽车和越野汽车以及液力传动的各类汽车取 1, *reason*;

η ——从发动机到主减速器从动齿轮之间的传动效率, *input*;

n ——计算驱动桥数, *input*;

(2) 按驱动轮打滑扭矩确定从动锥齿轮计算转矩

$$T_{Gs} = \frac{G_2 \cdot m_2' \cdot \varphi \cdot r_r}{i_m \eta_m}$$

式中 T_{Gs} ——计算转矩 $N \cdot m$;

G_2 ——满载状态下一个驱动桥上的静载荷 N , *input*;

m_2' ——汽车最大加速时的后轴负荷转移系数, *input*;

φ ——轮胎与路面间的附着系数, *reason*;

r_r ——车轮的滚动半径 m , *input*;

i_m ——主减速器从动齿轮到车轮之间的传动比, *input*;

η_m ——主减速器从动齿轮到车轮之间的传动效率, 当无轮边减速器时, 取 1, *input*;

总: 由(1), (2)计算的值取 T_{Ge} 、 T_{Gs} 较小者用作计算扭矩 T_G

(3) 按日常行驶平均(当量)转矩确定

$$T_{GF} = \frac{(G_a + G_{挂}) \cdot r_r}{i_m \eta_m n} (f_R + f_H + f_P)$$

式中 T_{GF} ——计算转矩 $N \cdot m$;

G_a ——汽车满载总重 N , *input*;

$G_{挂}$ ——所牵引的挂车的满载总重 N , *input*;

r_r ——车轮的滚动半径 m , *input*;

i_m ——主减速器从动齿轮到车轮之间的传动比, *input*;

η_m ——主减速器从动齿轮到车轮之间的传动效率, 当无轮边减速器时, 取 1, *input*;

n ——计算驱动桥数, *input*;

f_R ——道路滚动阻力系数, *reason*;

f_H ——汽车正常使用时的平均爬坡能力系数, *reason*;

f_P ——汽车或汽车列车的性能系数;

$$f_P = \frac{1}{100} \left(16 - \frac{0.195(G_a + G_{挂})}{M_{e\max}} \right), \text{ 当 } \frac{0.195(G_a + G_{挂})}{M_{e\max}} > 16 \text{ 时, 取 } f_P = 0;$$

3.4.3.2 主减速器齿轮基本参数的选择

(1) 主动齿轮的齿数 Z_1 , *reason*

(2) 从动齿轮的齿数 Z_2 , $Z_2 = \text{int}(Z_1 i_0 + 0.5)$; i_0 ——主减速器传动比, *fore*;

(3) d_2 , *reason*

$$d_2 = K_{d2} \sqrt[3]{T_G}$$

式中 d_2 ——(mm)从动锥齿轮的节圆直径 (初选), *reason*;

K_{d2} ——直径系数, 取 13~16, *input*;

T_G ——(N·m)取 T_{Gc} 、 T_{Gs} 较小者用作计算扭矩 T_G , *fore*;

(4) m

$$m = \frac{d_2}{Z_2}$$

式中 m ——(mm)从动锥齿轮大端端面模数;

d_2 ——(mm)从动锥齿轮的节圆直径, *fore*;

Z_2 ——从动齿轮齿数, *fore*;

$$\text{条件: } m = K_m \sqrt[3]{T_G}$$

式中 K_m ——模数系数, 取 0.3~0.4;

T_G ——(N·m)取 T_{Gc} 、 T_{Gs} 较小者用作计算扭矩 T_G , *fore*;

推理:

```
do{
    bool fin;
    for( $K_{d2}=13$ ;  $K_{d2} \leq 16$ ;  $K_{d2}++$ )
    {
        计算  $d_2(K_{d2})$ ; // 步骤 b
        if( $0.3 * \sqrt[3]{T_G} < d_2/Z_2 < 0.4 * \sqrt[3]{T_G}$ ) // 条件 1
        {
             $m = d_2/Z_2$ ;
            fin=true;
            break;
        }
    }
    步骤 a: //重新选择齿数
}while(!fin);
```

(5) 双级主减速器和普通圆柱齿轮式轮边减速器的圆柱齿轮副中心距 A 及齿宽 b 预选, 暂不考虑;

(6) 斜齿圆柱齿轮的螺旋角 β , 暂不考虑;

(7) 圆柱行星齿轮式轮边减速器, 暂不考虑;

(8) 螺旋锥齿轮与双曲面齿轮的从动齿轮齿面宽 F_2

$$F_2 = 0.155d_2$$

条件: $F_2 \leq 10m$

式中 F_2 ——(mm) 螺旋锥齿轮与双曲面齿轮的从动齿轮齿面宽, *reason*;

d_2 ——(mm)从动锥齿轮的节圆直径, *fore*;

m ——(mm)从动锥齿轮大端端面模数, *fore*;

(9) 双曲面小齿轮的偏移距 E , *reason*

轿车和轻型货车, 取 $E \leq 0.2d_2$; 其他货车和大客车, 取 $E \leq (0.1 \sim 0.2)d_2$

(10) 法向压力角 α , *reason*

轿车: $14^\circ 30'$ 或 16° ; 载重汽车: 20° ; 重型汽车: $22^\circ 30'$

(11) 螺旋方向。一般主动齿轮为左旋, 从动齿轮为右旋。

3.4.3.3 主减速器齿轮强度计算

(1) 单位尺长上的圆周力

① p_1 (N/mm)按发动机最大扭矩计算:

$$p_1 = \frac{2M_{e\max} \cdot i_g}{d_1 F} \times 10^3$$

式中 $M_{e\max}$ ——发动机最大扭矩 $N \cdot m$, *fore*;

i_g ——变速器传动比, 常取一档及直接档传动比, *input*;

d_1 ——(mm)主动锥齿轮的节圆直径 mZ_1 , m 、 Z_1 按前;

F ——(mm) 螺旋锥齿轮与双曲面齿轮的从动齿轮齿面宽, *fore*;

② p_2 (N/mm)按最大附作力矩计算:

$$p_1 = \frac{2 \cdot G_2 \cdot \varphi \cdot r_r}{d_2 \cdot F} \times 10^3$$

式中 G_2 ——满载状态下一个驱动桥上的静载荷 N , *fore*;

φ ——轮胎与路面间的附着系数, *fore*;

r_r ——车轮的滚动半径 m , *fore*;

d_2 ——(mm)从动锥齿轮的节圆直径 mZ_2 , m 、 Z_2 按前;

F ——(mm) 螺旋锥齿轮与双曲面齿轮的从动齿轮齿面宽, *fore*;

条件: $[p]$ ——(N/mm)许用单位齿长上的圆周力, *check*;

推理: 通过推导可知, 当不满足条件时, 只需调整 d_2 (通过调整 K_{d2}), 程序段表示如下:

```
int modified=0;
float p;
p=min(p1,p2);
Kd2=Val("Kd2");
while(p>[p])
```

```

{Kd2= Kd2+1;
If(Kd2>16){提示无法找到合适的值;}
p=min(p1,p2);
modified=1;}
if(modified= =1)
{writeval("Kd2");
return NULLVAR;}

```

(2) 轮齿的弯曲强度计算 $\sigma_w(\text{N/mm}^2)$

$$\sigma_w = \frac{2 \cdot T \cdot K_0 \cdot K_{sw} \cdot K_m}{K_v \cdot m^2 \cdot F \cdot Z \cdot J_w} \times 10^3$$

式中

T ——($\text{N} \cdot \text{m}$)该齿轮的计算扭矩;

K_0 ——超载系数, *fore*;

K_{sw} ——尺寸系数; 当端面模数 $m \geq 1.6 \text{ mm}$ 时:

$$K_s = \sqrt[4]{\frac{m}{25.4}}, \quad \text{当端面模数 } m < 1.6 \text{ mm 时取 } 0.5;$$

K_m ——载荷分派系数, 当两个齿轮均用骑马式支撑形式时, 1.00~1.10; 当一个齿轮用骑马式支撑时, 1.10~1.25。支撑刚度大时取小值,

reason;

K_v ——质量系数。一般取 1, *input*;;

m ——(mm)从动锥齿轮大端端面模数, *fore*;

F ——(mm) 计算的齿轮齿面宽, *fore*;

Z ——计算齿轮的齿数, *fore*;

J_w ——所讨论齿轮的轮齿弯曲应力的综合系数, *check*;

条件: 若是按 T_{Gc} 、 T_{Gs} 较小者计算 $\sigma_w \leq 700 \text{ N/mm}^2$;

若是按 T_{GF} 计算 $\sigma_w \leq 210.9 \text{ N/mm}^2$;

(3) 轮齿的接触强度计算 $\sigma_j(\text{N/mm}^2)$

$$\sigma_j = \frac{C_p}{d_1} \sqrt{\frac{2T_p \cdot K_0 \cdot K_{sj} \cdot K_m \cdot K_f \times 10^3}{K_v \cdot F \cdot J_j}}$$

式中

C_p ——材料的弹性系数, 对于钢制齿轮副取 $234 \text{ N}^{1/2}/\text{mm}$, *input*;

d_1 ——(mm)主动锥齿轮的节圆直径 mZ_1 , m 、 Z_1 按前;

T_p ——主动齿轮计算扭矩;

K_{sj} ——尺寸系数, 一般取 1, *input*;

K_f ——表面品质系数，一般取 1, *input*;

K_0 ——超载系数, *fore*;

K_m ——载荷分派系数，当两个齿轮均用骑马式支撑形式时，1.00~1.10;
当一个齿轮用骑马式支撑时，1.10~1.25。支撑刚度大时取小值，
fore;

K_v ——质量系数。一般取 1, *fore*;

F ——(mm)齿轮齿面宽，取齿轮副中较小值;

J_j ——齿面接触强度综合系数, *check*;

条件：若是按 T_{Ge} 、 T_{Gs} 较小者计算 $\sigma_j \leq 2800 \text{ N/mm}^2$;

若是按 T_{GF} 计算 $\sigma_j \leq 1750 \text{ N/mm}^2$ 。

3.4.3.4 主减速器齿轮的几何尺寸计算(略)

以上是主减速器设计计算所涉及的参数的提取分析过程。驱动桥知识约束提取全过程与此类似。

3.5 参数值更新算法的实现

前面我们已经详细阐述了数据库、尺寸约束函数库(简称函数库)和知识约束库(简称知识库)的功用和设计思路。简而言之，数据库存放的是所有的参数数据(包括参数值、参数名、参数间的关系等)，函数库存放的是尺寸参数的函数关系，知识库存放的是非尺寸参数计算所需的专业性知识(包括经验计算公式、描述性经验值等)。

本节要讨论的参数值更新算法将对数据库、函数库和知识库进行调用，它是三库的结合点。参数值更新算法所要解决的问题可以用一句话来概括：输入修改过的参数值，更新所有与之相关的参数值。

参数值更新算法是本软件系统智能化设计的关键所在，它被参数推理计算模块调用，并且是驱动桥的详细设计过程中参数修改所必需的算法。

3.5.1 建立数学模型

我们先来考虑这样的一个问题：假设有一参数集 $A\{a_1, a_2, \dots, a_n\}$ 和参数集 $B_i\{b_{i1}, b_{i2}, \dots, b_{im}\}$ ，它们之间存在如下函数关系

$$a_i = f(b_{i1}, b_{i2}, \dots, b_{im}) \quad (3-1)$$

其中 $0 < i < n+1$, i 为正整数

$$B_i\{b_{i1}, b_{i2}, \dots, b_{im}\} \subseteq A \quad (0 < m < n+1 \text{ 且 } a_i \notin B_i, m \text{ 为正整数})$$

即 A 中任意一个参数 a_i 与 A 中大于等于零个参数(不包括 a_i)存在函数关系。

式(3-1)中：

(1) 集合 B_i 中每一个元素都被称作 a_i 的源参数, a_i 是 B_i 中每一个元素的目的参数;

(2) 当 $B_i = \Phi$ 时，即 B_i 为空时，称 a_i 为无源参数;

(3) 若元素 $a \in A$, a 不是无源参数, 且 $a \notin B_1 \cup B_2 \cup \dots \cup B_n$, 则称 a 为叶结点参数;

(4) 若元素 a 是无源参数, 但不是叶结点参数, 则称参数 a 为入口参数;

(5) 若元素 a 既是无源参数, 又是叶结点参数, 则称参数 a 为孤立参数;

(6) 若元素 a 既不是无源参数, 又不是叶结点参数, 则称参数 a 为结点参数;

我们称符合上面条件的参数集为 A 类型参数集。

对于 A 类型参数集, 我们可以修改其中的任意几个元素, 由定义可知参数集中可能有元素会随之变化, 因此需要对参数集进行更新。按照修改的参数类型不同, 可以有以下几种情况:

(1) 只改动无源参数;

(2) 改动的参数中有叶结点参数或结点参数。

第二种情况比较复杂, 这里我们不对它进行讨论。对于第一种情况, 如果改变的参数全都是孤立参数, 那么参数集不需要更新, 因为孤立参数不会引起其它参数的变动。所以, 改动的参数中有入口参数的情况, 是我们着重分析的。

本软件系统数据库的参数表中所有参数构成的集合便是 A 类型参数集。因此, 当有部分参数发生改变时, 必须对相关参数进行更新。参数值更新算法的作用便在于此。

本系统中在两种情况下需要更新参数。

(1) 在初步设计时, 设计者输入原始参数后开始计算, 这时被修改的参数是所有的原始输入参数。这一功能由参数推理计算模块(参考图 2.2、2.5)来实现。

(2) 在详细设计时, 设计者对包括尺寸参数在内的参数进行修改。

我们首先分析参数推理计算模块的内部处理过程, 如图 3.9。

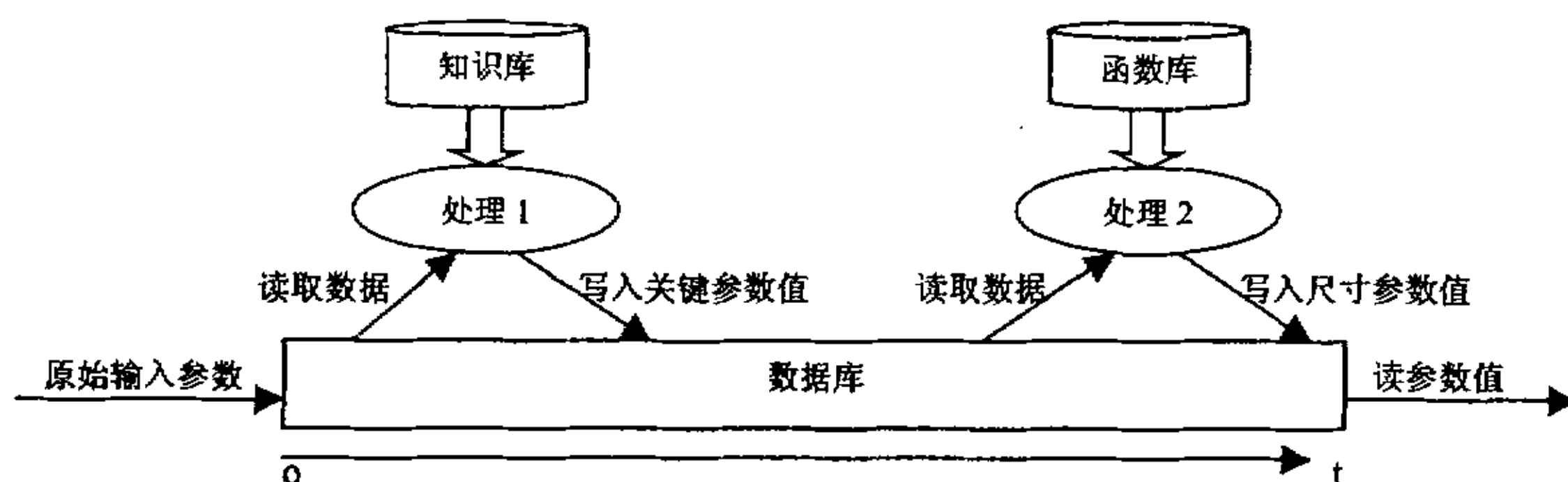


图 3.9 参数推理计算模块内部处理

图中从左到右时间递增: 最初用户输入原始参数值, 数据库记下参数值; 处理 1 读取数据库中的原始参数值, 并调用知识库的计算规则, 可以算出关键参数值, 同样记录到数据库中; 接着, 处理 2 读取数据库中的参数值, 并调用函数库中的函数, 结

合数据库中的参数关联表，可以计算出尺寸参数值，记录到数据库中以备调用。

可以看出，这个过程是参数值更新的一个特例。这里，所有被修改的数据都是原始输入参数(原始输入参数是无源参数)，通过调用参数值更新算法，可以计算出所有相关的参数。由于原始输入参数与所有参数直接或间接相关，所以不用去搜索与修改的参数相关的参数了。但是，必须知道计算参数值的顺序，即先计算哪个再计算哪个参数的问题。

另外，在驱动桥的详细设计时，用户可能需要对尺寸参数(注意：关键参数也属于尺寸参数)进行修改。由 A 类型参数集的定义可知，尺寸参数要么属于结点参数，要么属于叶结点参数。如果直接修改尺寸参数，是属于参数修改的第二种情况，为了将其转化为第一种情况，我们引入了尺寸变量参数(参考 3.2.2 节)。尺寸变量参数是入口参数，如果我们想要改变尺寸参数值，只要改变相应的尺寸变量参数即可。

当被修改的参数中包含了尺寸变量参数，处理过程就没前面那么简单了。因为，这时我们不能马上断定哪些参数会随着这些被修改过的参数变化的，就必须先搜索出因变参数，然后确定参数的计算过程进行计算。

通过上面的分析，我们知道一个完整的参数更新算法至少需要包含以下两部分：

- (1) 搜索所有由被修改的参数直接或间接引起改变的参数；
- (2) 确定搜索出的需要从新计算的参数的计算顺序。

3.5.2 算法解决思路

通过上面的分析，我们可以给出参数值更新算法的总体流程，如图 3.10。

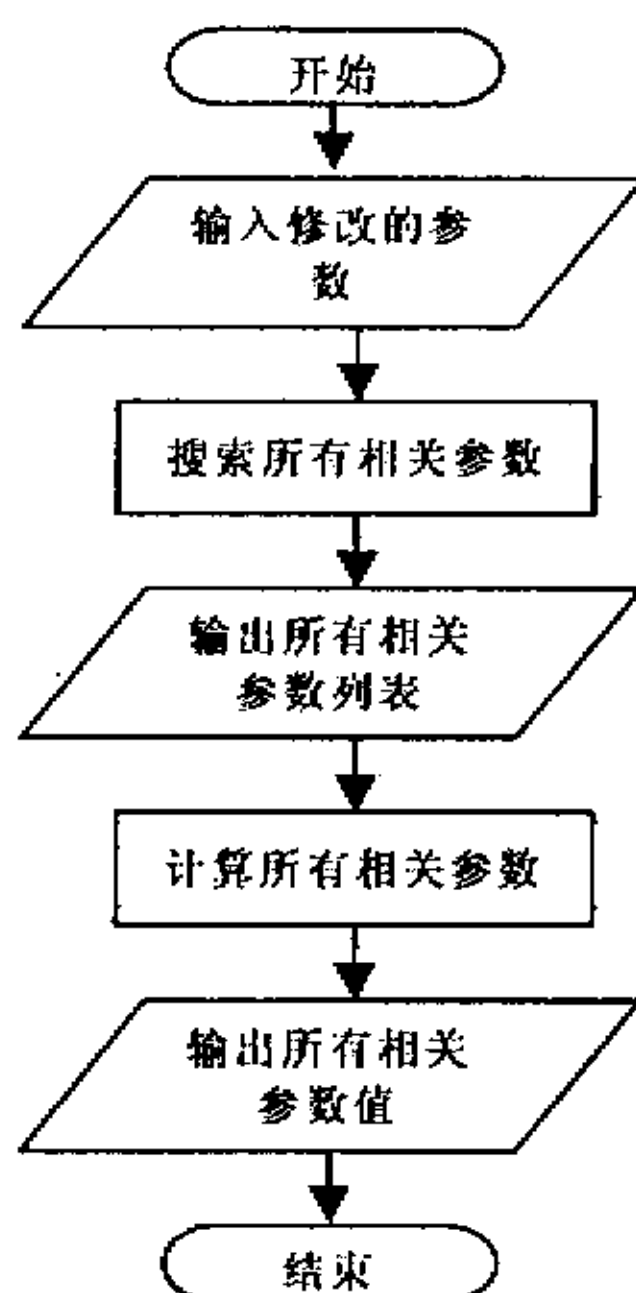


图 3.10 更新算法总体流程图

由总体流程图可以看出,算法主要解决两个子模块:搜索所有相关参数,计算所有相关参数。下面我们来分别解决这两个问题。

3.5.2.1 搜索算法

总体流程图中的搜索模块,输入的是被修改过的参数值,输出的是所有相关参数列表。

令可作为源参数的参数集合 $S = \{ s_1, s_2, \dots, s_x \}$, x 为集合 A 中可以作为源参数的参数个数。对于每一个源参数 s_i ($0 < i < x+1$), 有 s_i 的直接目的参数集合 $D_i = \{ d_{i1}, d_{i2}, \dots, d_{iy} \}$ 与之对应。因此,我们可以建立集合 A 的参数直接关联表(简称参数关联表,参考 3.2),如下图所示:

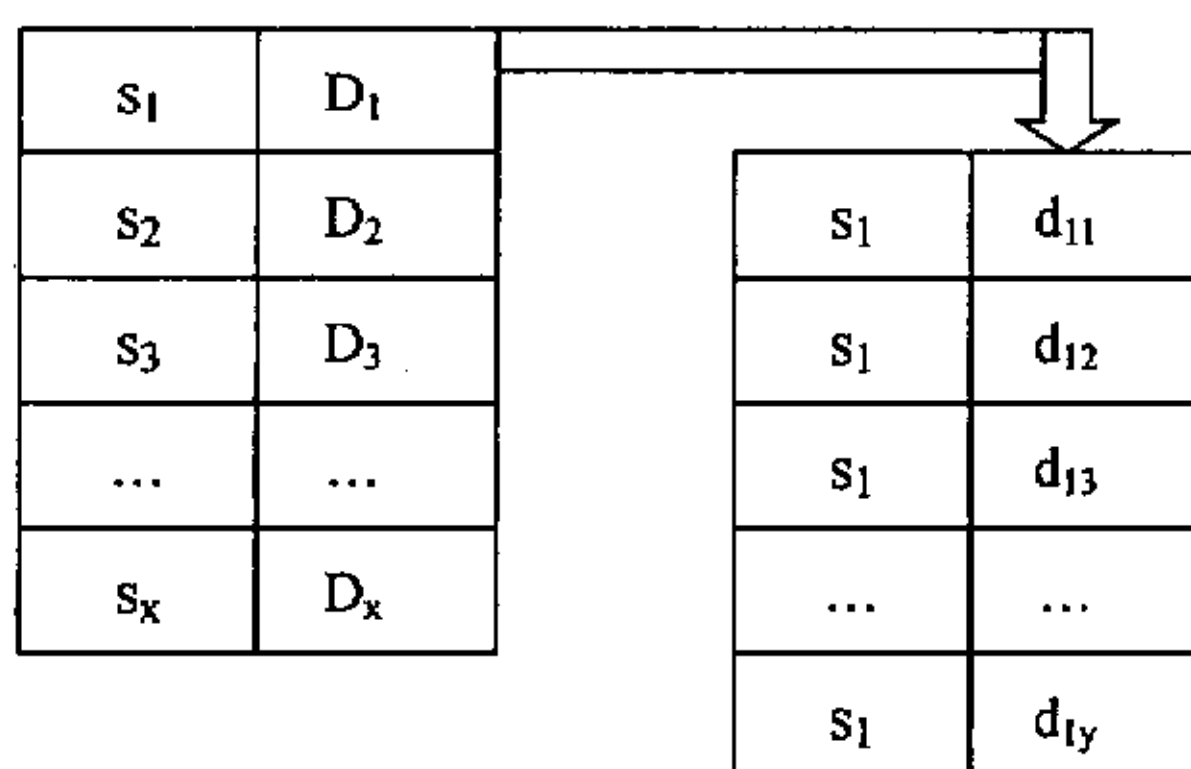


图 3.11 参数直接关联表

显然, $S = B_1 \cup B_2 \cup \dots \cup B_n$, $B_i (i \in [1, n])$ 的含义同前节, S 可以是入口参数或结点参数。集合 D 的生成过程可以用下面的步骤实现:

```

float A[n]={a1,a2,...,an};
float B[n][]={ {b11,b12,...,b1m}, {b21,b22,...,b2s}, ... };
n=集合 A 中元素个数;
for(int i=1;i<=n;i++)
{
    if(B[i]=  $\Phi$ ) continue;
    m=集合 B[i]中元素个数;
    for(int j=1;j<=m;j++)
        { 将 A[i]加入 B[i][j]的集合 D 中 }
}

```

集合 D 中可以是结点参数或则叶结点参数。

令设计者修改的参数集合 $M = \{ m_1, m_2, \dots, m_z \} \subseteq A$ ($0 < z < n+1$, z 为整数), 那么我们可以这样来搜索集合 M 所引起变化的参数:

- (1) 搜索 M 中每一个元素的直接目的参数, 产生一个无重复元素的集合 M_1 ;
- (2) 搜索 M_1 中每一个元素的直接目的参数, 产生一个无重复元素的集合 M_2 ;
- (3) 第 $i+1$ 次搜索是对第 i 次搜索产生的集合 M_i 进行搜索, 将搜索到的所有直接目的参数组合成一个无重复元素的集合 M_{i+1} ;
- (4) 在第 n 次搜索时, 若有 $M_n = \Phi$ 且 $M_{n-1} \neq \Phi$, 则搜索终止。

上面每次都是搜索直接目的参数, 因此每次搜索只需查找参数直接关联表就能实现。直接目的参数是相对于间接目的参数而言的, 例如, 有关系式: $a=b+c, d=a+e$ 。 b 的直接目的参数是 a , 即 b 的改变可以直接引起 a 的改变; b 的间接目的参数是 d , 因为 b 的改变直接引起 a 的改变, a 改变又直接引起 d 的改变, 因此 b 的改变间接地引起了 d 的改变。

通过以上的搜索步骤, 我们得到集合 M 的直接和非直接目的参数集合 $MD = M_1 \cup M_2 \cup \dots \cup M_n$, 集合 MD 就是与集合 M 相关的所有参数的集合。

但是, 我们必须注意到搜索在某些情况下有可能永无止境。搜索能终止的条件是: 搜索过程中不存在搜索回路。搜索回路的定义是这样的: 在搜索过程中, $\exists a \in A$, 对 a 按照上面的搜索算法进行搜索, 若 $\exists i > 0$, 使得第 i 步的搜索结果 M_i 中存在元素 a , 那么我们称此搜索过程存在搜索回路。如图 3.12 中存在搜索回路 $a1 \rightarrow a5 \rightarrow a2 \rightarrow a1$ 。图中有向边的起点表示源参数, 终点表示起点的直接目的参数。

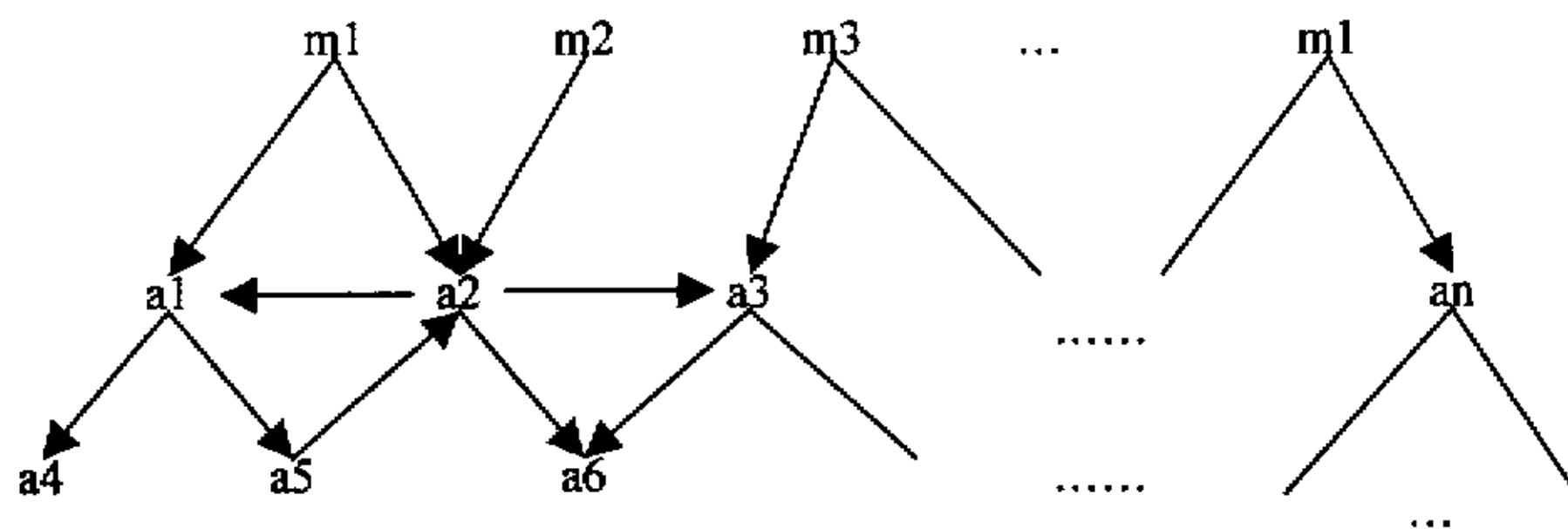


图 3.12 存在搜索回路的搜索图

3.5.2.2 参数更新

由于我们对参数的更新操作是建立在数据库表的基础上的, 因此前面提到的集合均以数据库表的形式来表示。如集合 MD 转换成表 MD , 每一步搜索得到的集合 M_i 转换成表 M_i 。

计算相关参数模块输入的是所有相关参数列表 MD，输出的是所有更新过的相关参数值。

相关参数的计算流程图如图 3.13。图中 CanCalculated 是一个布尔类型的参数，为 true 时表示可以对表 MD 当前记录表示的参数 PARA 进行计算；为 false 时表示参数 PARA 的直接源参数有未确定的，必须在 PARA 的直接源参数全部确定之后才能对 PARA 进行计算。

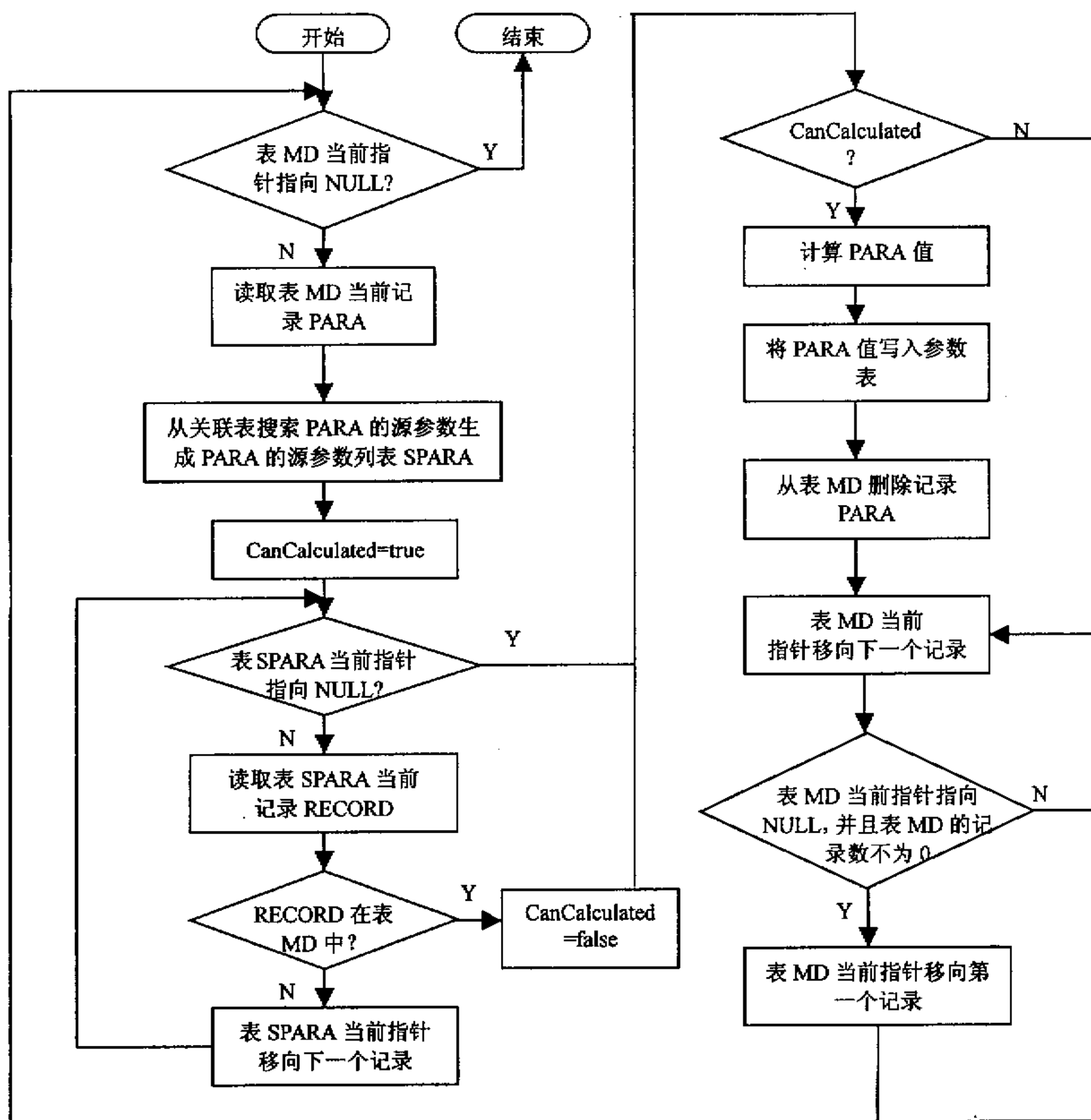


图 3.13 计算相关参数模块流程图

4 微型汽车驱动桥几何建模及分析

4.1 基于 CAD 集成软件的参数化设计方法简介

本文所用的 CAD 集成软件是 I-DEAS 软件系统。基于 I-DEAS 的参数化设计方法可归纳为两种方式：一种是基于程序库的参数化设计方法，一种是基于参数化图库的参数化设计方法。

4.1.1 基于程序库的参数化方法描述

基于程序库的参数化设计方法描述如下：将不同种类的标准件和常用件的几何参数分别存储于不同的数据文件中，运用开发工具编制若干具有独立功能的程序模块，用户输入少量的信息和参数后，启动相应的程序模块，自动完成数据访问、数据运算、构造编辑实体等操作，从而自动生成所需要的图形。该方法可用图 4.1 描述。



图 4.1 基于程序库的参数化方法示意图

该方法的实质是程序模块的设计和数据库的建立，各图形子程序模块由绘图编程语言编写。绘图编程语言是 I-DEAS 中重要的开发工具之一，它不仅提供了丰富的内部函数，而且在用绘图编程语言编写的程序中几乎可以调用全部的 I-DEAS 二维、三维绘图命令，因此绘图编程语言具有强大的图形处理功能^[19]。

但是完全用编程语言编写一个复杂的三维图形比较困难，通常在编写子模块时，先用 I-DEAS 绘制参数化图形作为基本图元，利用 FILE>>PROGRAM FILE>>CREATE 命令创建一个 PRG 文件录制绘制过程，在 PRG 文件的基础上修改得到子程序模块。但是该方法要将每一个参数都替换为子模块中的一个变量，而且对于复杂的、参数个数较多的三维图形，要将每个参数替换为一个变量而不出错依然是一个艰巨的任务。

4.1.2 基于参数化零件库的参数化方法描述

基于参数化零件库的参数化方法的思路为^[20]：I-DEAS 中有一个专用于零件系列化的模块，它可以将组件模型参数化，即提取组件模型的某些尺寸作为可以变化的参数，形成参数表，然后将参数化后的模型放入零件目录中，当要建立一个系列化的组件时，只需从零件目录中取出相应的参数化模型，然后在参数化表中填入合理数据即可得到想要的模型。也可以用批处理文件顺序读入数据库数据到参数化表中给参数赋

值,从而得到根据输入数据变化的三维模型,这种设计方法称为基于参数化零件库方法。该思路可用图 4.2 描述。



图 4.2 基于参数化图库的参数化方法示意图

这种设计方法有很多优点,本系统将采用这种设计思路。在 I-DEAS 系统中只要建立组件的参数化图库,将参数化表中的数据按同样的顺序存储在数据库中,系统只对数据库进行操作。当要根据设计数据显示三维模型时,将数据库中的数据写入顺序文件,然后用批处理文件将顺序文件数据赋值给参数化图库中的参数化模型,就可以建立系统的数据到 I-DEAS 参数化模型的传递,这种方法,使用者不需要掌握绘图编程语言,使用方便,适合对大批量组件的参数化设计。

零件目录(CATALOG)是 I-DEAS 中的一个用于专门存储参数化的组件模型的图库,本文将放在零件目录中的驱动桥零部件的参数化模型的集合称为参数化零件库。

4.2 微型汽车驱动桥参数化模型的建立

4.2.1 微型汽车驱动桥结构分析及其离散化

微型汽车驱动桥组件层次是按照其装配关系形成树状结构,层次图中的总结结点由 CParts 类描述,其它结点由 CPart 类描述,CParts 类和 CPart 类具体设计见详细设计的数据库设计。对于三段式微型车驱动桥最顶层结构如图 4.3 所示,驱动桥总成由两个主要部分,桥壳总成和主减速器总成。另外,还有若干其它组件,如图 4.4 所示。

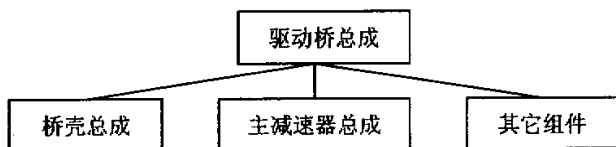


图 4.3 驱动桥总成结构

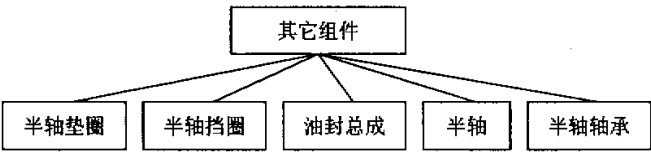


图 4.4 驱动桥总成中的其他组件

桥壳总成如图 4.5 所示，主减速器总成如图 4.6 所示，在主减速器总成中还包括差速器总成，它的组成如图 4.7。

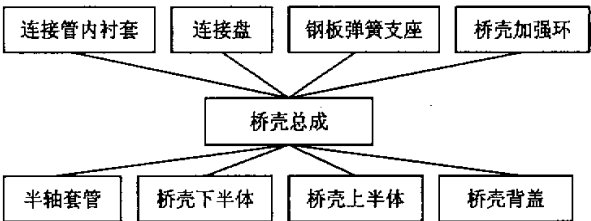


图 4.5 桥壳总成结构

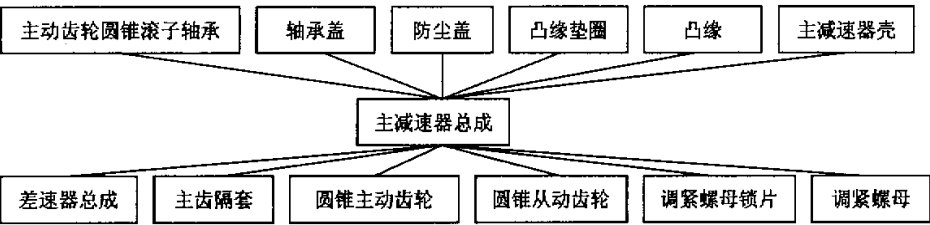


图 4.6 主减速器总成结构

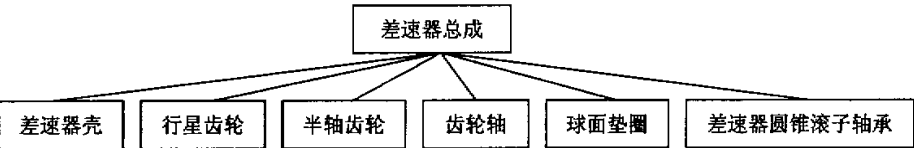


图 4.7 差速器总成

按照驱动桥组件层次的划分，对所设计的组件进行分类，然后建立每种组件结构类型的模板，称之为组件模板，这些模板反映了该系列现有常用组件中可能出现的结构，同时考虑有些结构在某个具体的组件中可能退化消失。随着参数的改变，组件模板的图形会出现三种可能的变化^[21]：第一种：图形只有尺寸变化，结构保持不变。第二种：由于尺寸的变化而导致结构变化，如圆锥变为圆柱。第三种：结构随参数而变化。

在驱动桥智能化设计系统中,以某系列四种典型驱动桥为主模型,配合其它相似结构类型的驱动桥,对其进行离散,提取典型特征。离散的原则为^[22]:

- (1) 近似组件提取相同的特征。
- (2) 特殊组件单独进行特征提取。
- (3) 进行特征提取时,应尽可能抽象其相似特征。
- (4) 特征提取时,应考虑其实现的可能性。

图 4.8、4.9 分别为预先建立的差速器壳体 and 桥壳背盖的特征几何模型。

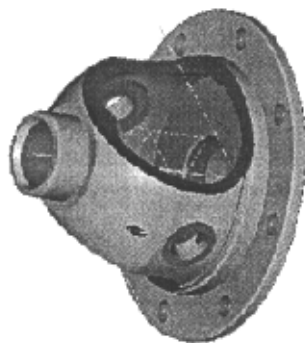


图 4.8 差速器壳体特征几何模型



图 4.9 桥壳背盖特征几何模型

4.2.2 组件数据定义

组件数据分为描述组件几何形状的几何数据以及描述组件特性的非几何数据,包括与装配相关的数据和与组件相关的数据。实际上,在进行组件参数化建模时,一般应用以下几类数据^[24]:

- (1) 基本几何要素:表示组件的几何形状的点、线、面信息;
- (2) 拓扑关联信息:表示基本几何要素之间的联接关系;
- (3) 形状要素:表示组件形状特征的信息,诸如孔、槽、凸台、型腔等;
- (4) 变量信息:组件结构的允许变动量;
- (5) 装配信息:组件装配间的关系,配合类型及联接信息等;
- (6) 功能信息:主要指材料、热处理、生产过程和管理数据等;^[23]

组件数据有些是在组件建模时用到,有些是在装配时用到,有些则是在后处理时才用到,但是在开始建模就必须予以充分的考虑。

4.2.3 组件的参数化模型建立

组件参数化模型的建立过程也就是确定组件数据的过程,本文在建立参数化模型时,遵循以下的步骤和原则。

- (1) 驱动桥离散化组件模型描述

按照驱动桥的功能特征进行离散化后的每一类组件依然数量繁多、结构各异,必

须对某一类的组件进行分析,抽象出运用实体造型方法构造组件时所用的特征类型,将每一类的组件抽象为少数几个模型;在提取某些组件特征进行抽象时,依据所提取的共同特征形成组件模型。考虑到形成过程中,实际制作的方法不同,将形成几个组件模型,调用时选择不同的组件模型。

(2) 组件模型的特征构形描述

运用 I-DEAS 的实体造型工具产生所需要的驱动桥组件模型,同一个组件模型可以采用不同的造型方法,造型的步骤以及造型用的参数也可以不同。组件模型的构形过程的描述包括造型操作的步骤、每一步造型操作的参与特征与特征引用位置,只有按照实际生产制造过程的特征构形过程产生的数据才比较具有实用价值。

(3) 组件尺寸参数的定义与关系描述

工程实践中具有尺寸系列的常用零件,其尺寸参数可分为主驱动参数和附驱动参数,一般只需给定主驱动参数便可唯一确定零件,但从纯几何角度来讲,只有主驱动参数是不够的,还必须具有一些可以由主驱动参数确定的附驱动参数。附驱动尺寸参数与主驱动尺寸参数的关系提供三种描述方法^[25]:①表格方法 ②公式方法 ③表格与公式结合。对于没有严格意义上系列化的零件模型,也可以确定主驱动尺寸参数和附驱动尺寸参数,用主驱动尺寸参数约束附驱动尺寸参数的选值范围,以保证零件模型的拓扑结构,力求零件模型合理,又要使模型的尺寸参数尽可能少。

4.3 受力件的强度有限元分析

驱动桥是汽车总成中的主要承载件之一,其强度的大小将直接影响汽车的有效使用寿命,局部应力集中将导致桥壳的局部开裂甚至断裂,为了使驱动桥能够承受在高速以及不平路面行驶时产生的动载荷,可以通过对驱动桥桥壳拟采用有限元方法对应力分布形态进行分析,并在此基础上提出改进措施。下面以由本建模系统形成的某微型车驱动桥为例,来说明如何对桥壳进行有限元分析。

4.3.1 微型汽车驱动桥结构特点

该微型车后悬架系统采用螺旋弹簧作为弹性元件,弹簧下端支撑于桥壳两端的弹簧支座上;筒式减振器下端支撑位于驱动桥后侧的减振器支座上;纵向推力杆支架位于弹簧座与轮毂接头之间,下图为该驱动桥壳一侧的结构简图^[26]。

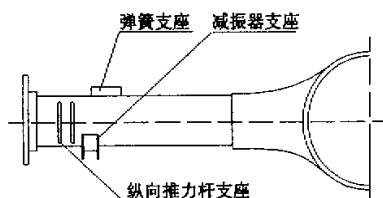


图 4.10 微型汽车驱动桥结构示意图

4.3.2 微型汽车驱动桥壳有限元模型的建立

(1) 创建桥壳三维几何模型

三维实体几何模型是建立有限元模型的基础,本文采用集成工程设计分析软件包 I-Deas 作为桥壳三维建模平台,利用其中 Master Modeler 模块创建桥壳三维几何模型,如图 4.11。



图 4.11 桥壳三维几何模型

(2) 桥壳的受力分析

根据驱动桥实际承载情况,驱动桥所受载荷包括下列两类:簧载质量,纵向推力杆的反作用力。

该微型车在满载时的后悬架载质量为 940kg,驱动桥每一侧为 470kg,根据悬架与驱动桥的连接方式,本文取驱动桥每一侧的静载荷沿弹簧支座均匀分布,并施加在相应的节点上,簧载质量作用形式如下图 4.12 中 f 所示。

汽车驱动力通过车轮、驱动桥、纵向推力杆传到车身,推动车身前行,因此驱动桥壳体还受到纵向推力杆的反作用力 p 的作用。反作用力在桥壳上的作用形式如图 4.12 中 p 所示。

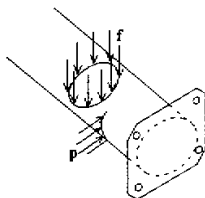


图 4.12 桥壳载荷的施加形式

本文在对驱动桥壳体进行有限元分析时,计算工况取极端形式:即汽车满载,在不平路面上行驶时紧急制动。

汽车处于这种工况,受两类动载作用,即竖直动载荷以及纵向动载荷。垂直动载荷取后桥满载负荷的 2.5 倍,即取动荷系数为 2.5,因此桥壳每一侧应施加垂直动载荷为:

$$F = \sum f = \frac{1}{2} m_2 g \times 2.5$$

$$=470 \times 9.8 \times 2.5 = 11515 \text{ (N)}$$

上式中, m_2 为驱动桥簧载质量。

水平动载荷由纵向推力杆提供, 其值取后单轮所受最大制动力 F_r 大小的 1.5 倍, 即:

$$\begin{aligned} P &= \sum p = 1.5 F_r \\ &= 1.5 \times 11910 = 17865 \text{ (N)} \end{aligned}$$

上式中 F_r 的大小来源于参考文献[27]。

(3) 网格划分

在用有限元法求解弹性力学问题时, 单元划分的好坏与计算结果的精确度及计算消耗的机时有很大的关系, 因此, 划分单元是进行有限元分析很重要的一环。

通常情况下, 应力的计算误差与单元的尺寸成正比, 位移的误差与单元尺寸的平方成正比, 单元划分愈大, 位移的误差愈显著^[28]。虽然从理论上说, 单元分得越小, 网格划得越密, 计算结果越精确, 但随之而来的是计算量越大, 计算时间越长, 对计算机存储量的要求也越大。计算表明, 当加密网格超过一定的限度, 不但不能提高精度, 有时反而使精度降低, 因为过分地加密网格, 使得计算量激增, 从而导致计算误差的增大, 所以要根据对精度的要求, 计算机容量及合理的计算时间来确定单元的大小。

利用 I-DEAS 网格形成模块 (Meshing) 划分网格分两个步骤: 第一步确定单元的大小和类型参数, 第二步是在几何实体上建立网格。

在驱动桥有限元分析中根据不同的计算目的, 可选用不同的单元模型, 通常采用板单元划分桥壳, 微型车驱动桥中采用四节点单元已能满足精度要求, 因此本文采用四节点板单元对微型汽车驱动桥壳体两端的几何模型进行网格划分, 而中间较薄的主减速器壳体用薄壳单元划分, 因此驱动桥壳体有限元模型中共含 3223 个薄壳单元, 1592 个板单元。在网格划分过程中, 选取桥壳材料物理属性如下表 4.1。

材料规格	密度 (kg/m^3)	泊松比	切变弹性模量 (GPa)	杨氏弹性模量(GPa)
钢	7900	0.3	79.4	206

表 4.1 桥壳材料物理属性

(4) 加约束和载荷

约束桥壳两端所有节点的六个自由度, 并按照前面的受力分析结果, 将约束和载荷加到划分好网格的模型中, 形成最终的桥壳有限元模型, 如图 4.13。

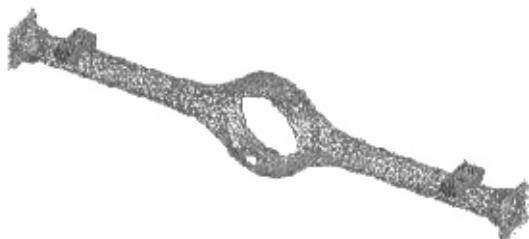


图 4.13 驱动桥壳有限元分析模型

4.3.3 有限元求解结果分析

I-DEAS 模型求解模块 (MODEL SOLUTION) 是该软件包中的一个集成部分, 分析时不需要其它的附加输入文件, 只要把有限元分析模型提交给模型求解器, 就可自动求解。解算前节的有限元模型得到桥壳应力图和应变图如下:

Figure 4.14: Stress distribution cloud map of the bridge housing. The image shows a 3D model of the housing with a color-coded stress distribution. A vertical color bar on the right indicates the stress scale, ranging from 0 to 845 MPa. The highest stress is concentrated at the mounting points.

图 4.14 桥壳应力分布云图

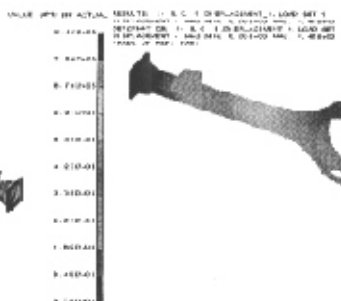


图 4.15 桥壳变形云图

由上述解算结果可以看出, 在桥壳两侧厚度为 3.2mm, 半轴套管壁厚 3.5mm 时, 桥壳最大应力为 845Mpa, 而桥壳材料的屈服极限 $\sigma_s=195\text{MPa}$, 强度极限 $\sigma_b=330\text{MPa}$, 因此桥壳最大应力大于材料的屈服应力值, 在重新设计时, 有必要改变该处的受力形态, 以降低该处最大应力值。

5 微型汽车驱动桥智能化设计系统的集成方法

前面几章我们讨论了驱动桥智能化设计系统的内部主要模块的实现；又讨论了零件库模型的建立；另外，还提到 I-DEAS 软件系统在驱动桥智能化设计系统中的功能。即如图 2.1 所示的三个有一定独立性的模块，它们各自可以完成独立的功能，可以独立运行。例如，可以往零件库中加零件，而设计系统和 I-DEAS 系统不需要对其进行控制；设计系统可以自行运行，并输出各类参数结果，而不管 I-DEAS 系统和零件库存在与否。但是，要想完成一个特定的功能：显示并分析最终的设计结果，就必须考虑如何将它们集成到一块儿工作。

如何让这三部分协调工作，是我们这一章讨论的中心问题。

5.1 I-DEAS 的宏命令和高级语言简介

驱动桥智能化设计软件系统图形处理是在 I-DEAS 软件系统的内部，采用 I-DEAS 软件的宏语言和其高级语言实现^[34]。所谓的宏，就是一组 I-DEAS 的命令的集合，可完成一定的功能。而 I-DEAS 提供的高级语言（PRG 语言）是一种类 FORTRAN 语言，具有与 FORTRAN 类似的编程特性。

PRG 语言的格式为：

标识符	空格	冒号	语句或注解
-----	----	----	-------

常用的标识符为：K，语句标识；C，注释标识；E，结束符。

语句分为宏和 PRG 语言。宏是由菜单命令的缩写构成；程序语言是由类 FORTRAN 语言构成。宏可由 I-DEAS 软件系统直接生成，故其较为简单。

I-DEAS 软件系统的高级语言包括：

(1) 数据定义：I-DEAS 软件系统的数据类型分为整数、浮点数、字符串。定义变量可使用 DECLARE 语句或直接使用。

```
# DECLARE PART (5)
# PI=3.14159
# STR= "SAMPLE"
```

(2) 输入、输出函数： INPUT、OUTPUT

```
# INPUT "请输入 S 值" S
# OUTPUT "SAMPLE"
# OUTPUT S ; 将 S 值输出;
```

(3) 循环控制语句： IF……THEN…… , GOTO;

```

# BEGIN:
.....
# IF (I NE 5) THEN GOTO BEGIN
# GOTO NEXT
.....
# NEXT:
.....

```

(4) 文件读写: OPEN、READ、WRITE、CLOSE

OPEN 为打开一个文件, READ 为从一个文件中读数据, WRITE 为向一个文件中写数据, CLOSE 为关闭一个打开的文件。

```

# OPEN FP "TEST.DAT"
# 从 FP 中以格式 (F3.0) 读取一浮点数到 PARTNUM
# READ FP FORMAT "(F3.0)" PARTNUM
# 从 FP 中读取一行字符串到 STRING
# READ FP STRING
# 关闭文件 FP
# CLOSE FP

```

(5) 逻辑判断: AND、OR、NOT、EQV 等。

(6) 出错处理:

ON_ERROR 是应用程序用来检测错误, 该命令有三种选择, 即 IGNORE、PAUSE、local handler。其中 IGNORE 是忽略程序中的当前的错误使程序继续执行, PAUSE 是错误一旦被检测出来便终止程序, local handler 是转向错误处理程序。

5.2 零件库的初始化

驱动桥智能化设计系统的零件模型是存放在两个独立的文件 axle.mf1 和 axle.mf2 中的, 因此若想在 I-DEAS 系统中调用, 必须将零件模型转化到 I-DEAS 系统的参数化零件库中。零件初始化可以在安装驱动桥智能化设计系统时进行, 也可第一次运行系统时进行, 但必须保证初始化前 I-DEAS 系统已经正确安装。初始化的实现主要用 I-DEAS 系统自带的宏命令和高级语言来完成, 本系统中为 catalog.prg 文件。

catalog.prg 实现框架如下:

```

K : Declare partNum //定义变量partNum
K : partNum=1 //初始化为1
K : $ mpos :: /ma pu //按put away...按钮

```

```

K : ALL //右键后选All
K : $ return
K : $ mpos ;; /MA CA //进入零件目录管理对话框
K : CC //按Create Catalog...按钮, 进入创建零件目录对话框。
K : PR Shared //填写Project栏为: Shared
K : CA microaxle //填写Catalog栏为: microaxle
K : OKAY //退出创建零件目录对话框
K : OKAY //退出零件目录管理对话框
K : $ return
K : Begin //进入循环
K : $ mpos ;; /ma ma //选择Manage Bin...命令
K : ET partNum; //选择第partNum个零件
K : G //取出该零件
K : $ return
//选择Modify Catalogs...命令, 进入Catalog对话框
K : $ mpos ;; /o cs gdb_Catalog_subpanel; 1202; D; 1; D;
K : $ return
K : $ /ca mc pa //选择Parameter命令
//选择零件
BP: 1 1 -3.27236891E-01 -2.58053660E-01 428407 0
BR: 1 1 -3.27236891E-01 -2.58053660E-01 428618 256
K : OKAY
K : $ /ca mc ch //选择Check in命令
BP: 1 1 -2.30401516E-01 -4.91794348E-01 437510 0
BR: 1 1 -2.30401516E-01 -4.91794348E-01 437711 256
K : CP //在弹出菜单中选择Catalog Part, 弹出Catalog对话框
K : PN Shared //填写Project栏为: Shared
K : L microaxle //填写Part Catalog栏为: microaxle
K : OKAY
K : OKAY
K : $ return
K : partNum=partNum+1 //标记下一个零件
K : IF (partNum NE 40) THEN GOTO Begin //判断是否退出循环
K : $ mpos ;; /F PR E //结束

```

5.3 系统的接口设计

在概要设计中，我们已经提过接口在本软件系统整体构架中的位置，如图 2.1。接口设计主要是实现本软件系统与 I-DEAS 系统的数据通信。为了便于分析，我们将图 2.1 中的 I-DEAS 系统单独提取出来，分析它的输入与输出数据流，如图 5.1。

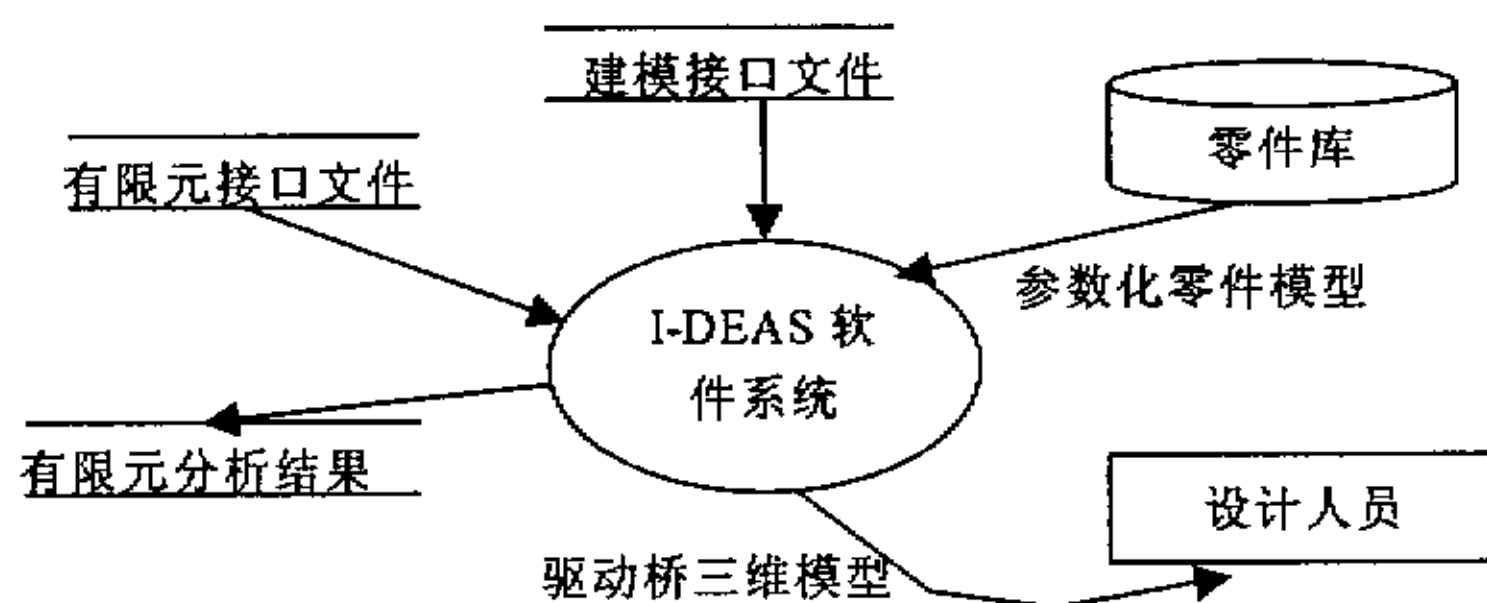


图 5.1 I-DEAS 系统的输入输出数据流

由图 5.1 可见，I-DEAS 系统主要对以下三方面的数据进行处理：建模接口文件，有限元接口文件，零件库。零件库是从 I-DEAS 系统内部建立的，因此，不需考虑与 I-DEAS 系统接口的问题。建模接口文件与有限元接口文件是由驱动桥智能化设计系统根据设计结果生成的，而它们是 I-DEAS 系统进行建模和有限元分析的唯一数据来源。也就是说，驱动桥智能化设计系统与 I-DEAS 系统的数据通信完全依赖于这两个接口文件。因此，建模接口文件与有限元接口文件的设计好坏，将直接影响到设计结果在 I-DEAS 系统中处理分析的效率高低、正确与否。

下面我们将从接口文件的设计、接口文件生成模块的实现和接口文件的调用方法三方面来详细讨论接口设计是如何实现的。

5.3.1 接口文件的设计

我们主要讨论建模接口文件的设计，有限元接口文件原理相同。建模接口文件主要是用来在设计系统和 I-DEAS 系统之间传递数据信息的。而且数据信息是从设计系统传向 I-DEAS 系统，可见建模接口文件是由设计系统产生并写入数据而由 I-DEAS 系统读取处理的。因为 I-DEAS 系统读取数据的方式是给定的，所以问题的关键是如何设计一种 I-DEAS 系统能够正确读取的数据记录格式。

零件库中的每个零件都有一个唯一编号，而且起始编号为 1。I-DEAS 系统读取尺寸信息后就是按照此编号来赋值的，即按零件编号从小到大的顺序 1、2、3.....，对每个零件的尺寸进行赋值^[35]。

建模接口文件所要记录的数据主要是零件的尺寸值。除此之外，若要正确读取数

据还需要一些辅助信息, 包括: 总零件数, 每个零件在零件库中的编号, 每个零件由多少个尺寸。因此, 可以考虑如下设计方案:

表头;

段 1;

段 2;

.

.

.

END

其中表头说明该文件主要信息, 如零件的数目; 段主要描述一个零件的信息(包含零件标识符、零件参数数目、零件参数值); END 为结束符。

实例如下:

表头	→	7	// 零件数目为 7 个
段 1	{	1	// 零件标识为 1
		3	// 零件参数数目为 3 个
		10.0000	// 第一个参数值
		9.0000	// 第二个参数值
		23.0000	// 第三个参数值
段 2	{	8	// 零件标识为 1
		6	// 零件参数数目为 6 个
		20.5000	// 第一个参数值
		7.0000	// 第二个参数值
		23.0000	// 第三个参数值
		.	
		14.0000	// 第六个参数值
		
结束符	→	END	

5.3.2 接口文件生成模块的实现

接口文件的数据主要来源于数据库中的零部件表和参数表。因此要生成接口文件必须对数据库中零部件表和参数表的内部数据结构非常了解, 表的数据结构请参考详细设计中的“数据库设计”一节。对于数据库表的查询, 我们利用在源程序中嵌入结构化查询语言 (Structured Query Language, SQL) 来实现^[36]。生成接口文件的程序流程图如图 5.2。

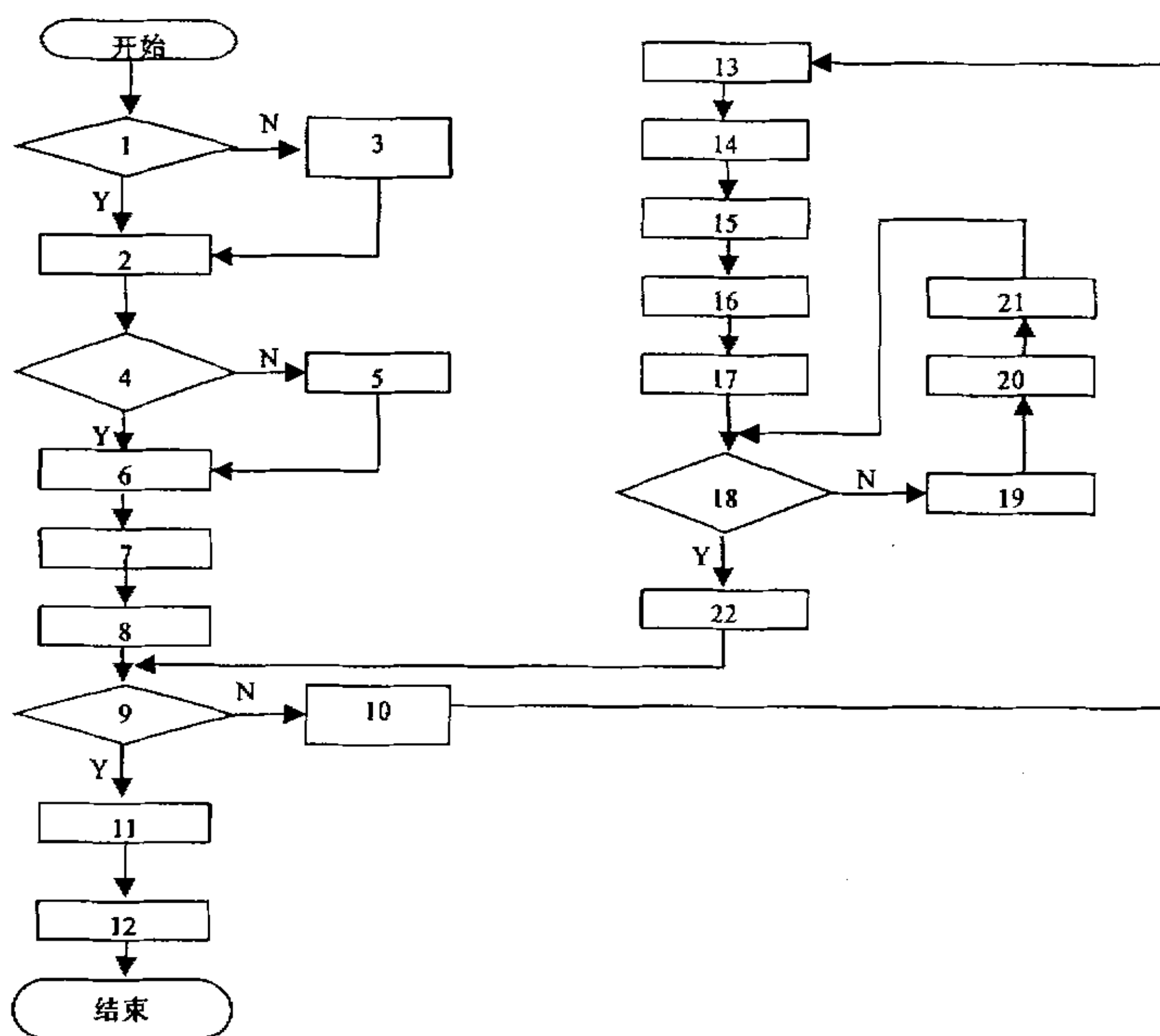


图 5.2 生成接口文件流程图

流程图说明:

1. 判断建模接口文件(model. dat)是否存在;
2. 以写方式打开 model. dat;
3. 生成一个空的 model. dat;
4. 数据库是否已打开;
5. 打开数据库文件;
6. 用结构化查询语言(Structured Query Language, SQL)生成临时表 Tpart:
 SELECT * FROM 零部件表 WHERE 类型= 'part' ORDER BY 零件库 ID
7. 计算表 Tpart 中的零件个数 partnum;
8. 将 partnum 写入 model. dat;
9. 判断表 Tpart 的记录指针是否为 NULL, 即判断记录指针是否指向表尾;
10. 读取表 Tpart 中当前记录的“零件库 ID”字段值;
11. 关闭数据库文件;

12. 关闭 model.dat;
13. 将读取的“零件库 ID”字段值写入 model.dat;
14. 读取表 Tpart 中当前记录的“尺寸数”字段值;
15. 将读取的“尺寸数”字段值写入 model.dat;
16. 读取表 Tpart 中当前记录的“所属部件 ID”字段值 partID;
17. 生成临时表 Tdimension, SQL 语句如下: SELECT 参数 ID, 参数值 FROM 参数表 WHERE (所属零件 ID=partID AND 参数类型=‘尺寸参数’) ORDER BY 参数 ID;
18. 判断表 Tdimension 的记录指针是否为 NULL;
19. 读取表 Tdimension 中当前记录的“参数值”字段值;
20. 将读取的“参数值”字段值写入 model.dat;
21. 将表 Tdimension 的记录指针移向下一个记录;
22. 将表 Tpart 的记录指针移向下一个记录。

5.3.3 接口文件的调用方法

接口文件主要是被 I-DEAS 系统调用的。I-DEAS 系统通过运行 NewStartUP.PRG 文件来实现对接口文件的调用。NewStartUP.PRG 的整体框架如下:

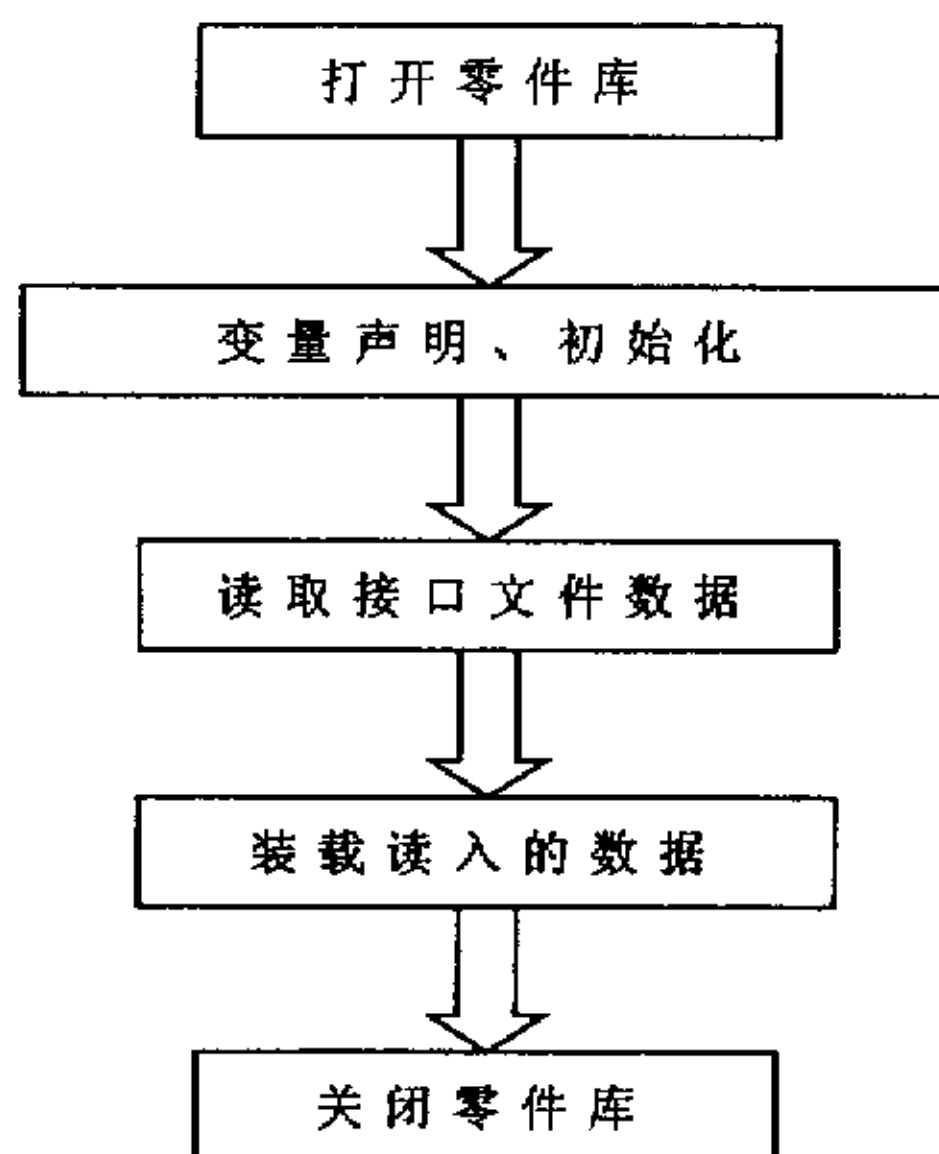


图 5.3 NewStartUP 整体框架示意图

其中，读取接口文件数据模块是核心，它的程序流程图如下:

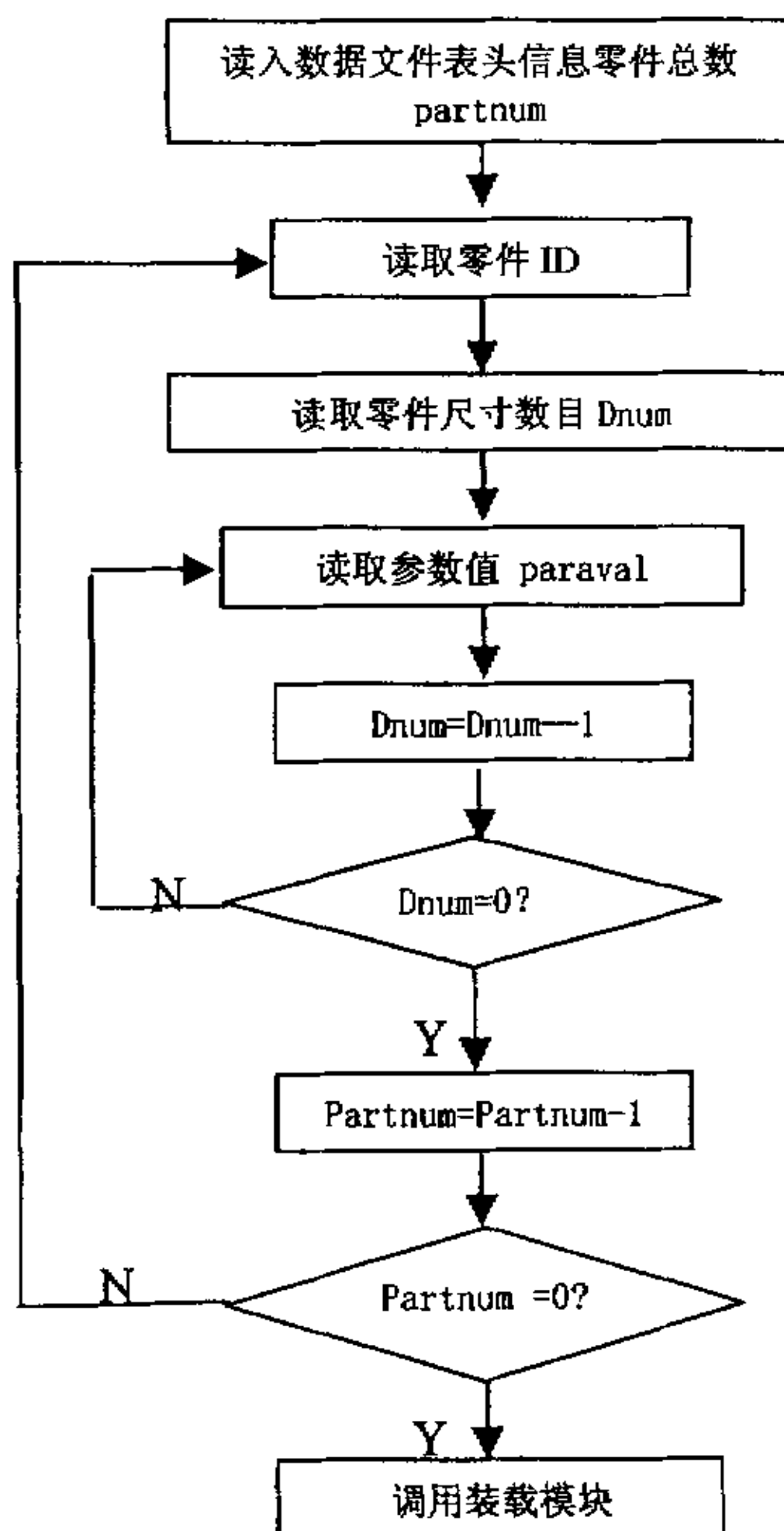


图 5.4 读取接口文件数据流程图

NewStartUP.PRG 的源程序详见附件。

5.4 系统的集成过程

至此，驱动桥智能化设计系统的各个子模块及接口技术设计基本完成，各个模块的调用过程如图 5.5 所示。

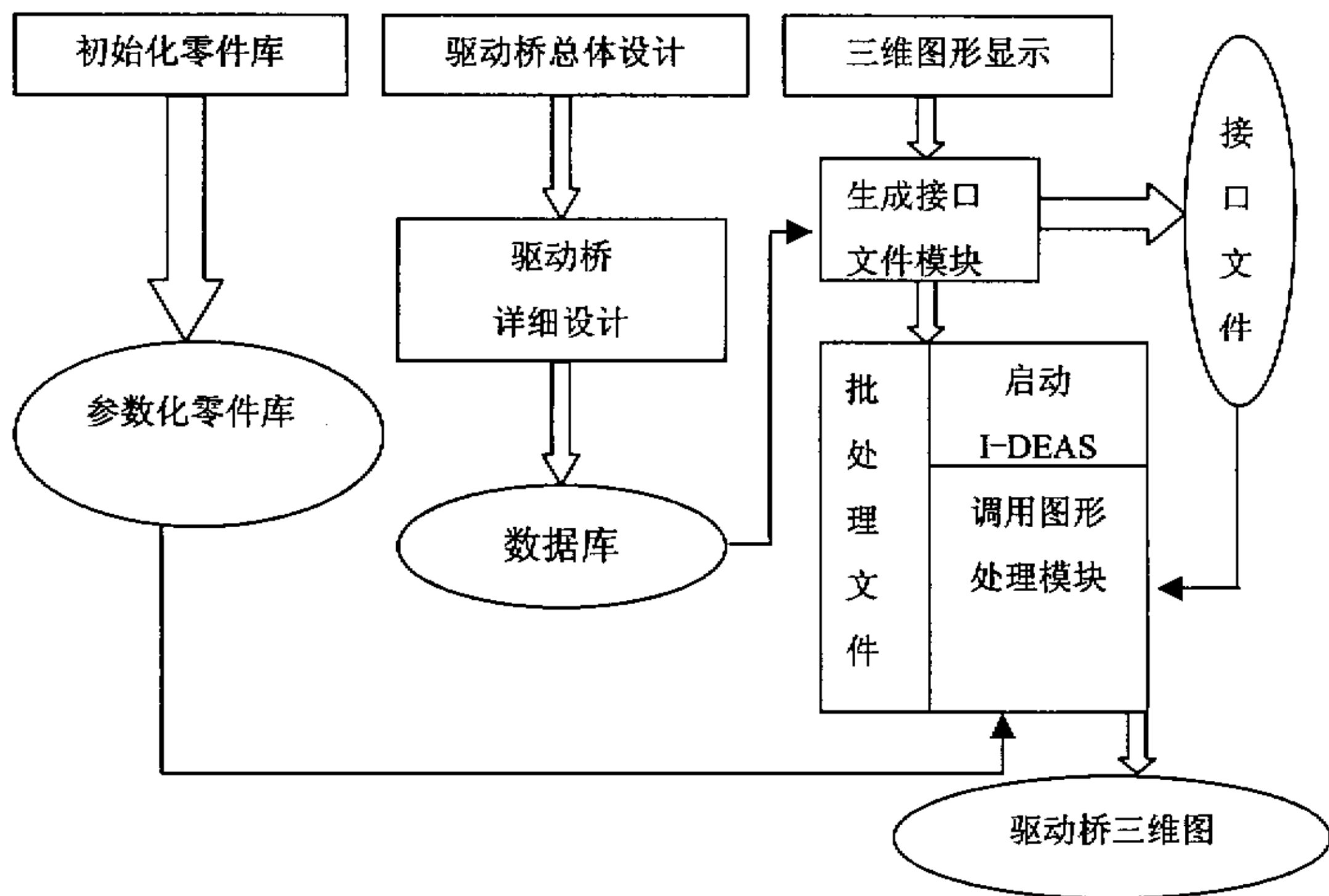


图 5.5 系统集成过程图

由上图可以看出，本系统集成的核心在于三维图形显示模块。初始化图形库在一个操作系统下只需运行一次；在驱动桥的总体设计和详细设计中，得到保存有该驱动桥所有组件的全部参数的数据库，然后调用三维图形显示模块显示所设计的驱动桥。三维图形显示模块先调用接口文件生成模块生成接口文件，然后启动一个命令脚本文件，该命令脚本文件的作用是启动 I-DEAS 和图形处理模块。

前面几节中，介绍了系统集成所必需的几个文件：用于初始化零件库的文件 catalog.prg，用于数据通信的接口文件 model.dat，用于 I-DEAS 系统调用接口文件的 NewStartUP.PRG。现在所缺的是如何对它们进行控制，我们用两个命令脚本文件 myideascata.cmd 和 myideasnew.cmd 来实现。而命令脚本文件采用调用一个进程的方法来启动，在 VC 中可以用以下命令完成^[40]：

```
ShellExecute(NULL,NULL,_T("myideascata.cmd"),NULL,文件所在路径,SW_SHOW);
```

```
ShellExecute(NULL,NULL,_T("myideasnew.cmd"),NULL,文件所在路径,SW_SHOW);
```

文件之间的调用关系如图 5.6。

命令脚本文件 myideasnew.cmd 如下：

```
echo off //不显示启动过程的命令
```

```
call setup_varbs.cmd 所在路径 //设置环境变量，路径在初始化时获得
```

```
%SDRC_INSTL%\ideas\ideas.exe -d ogl -e /f pr r fil newstartup.prg 所在路径 okay%*
```

//系统接口利用 I-DEAS 的启动参数来实现参数文件的传递。

命令脚本文件 myideascata.cmd 如下:

echo off //不显示启动过程的命令

call setup_varbs.cmd 所在路径 //设置环境变量, 路径在初始化时获得

%SDRC_INSTL%\ideas\ideas.exe -d ogl -q -g -aDesign -tMaster Modeler -pAdministrator -m
文件 axle.mfl 所在路径 -e /f pr r fil 文件 catalog.prg 所在路径 okay%*

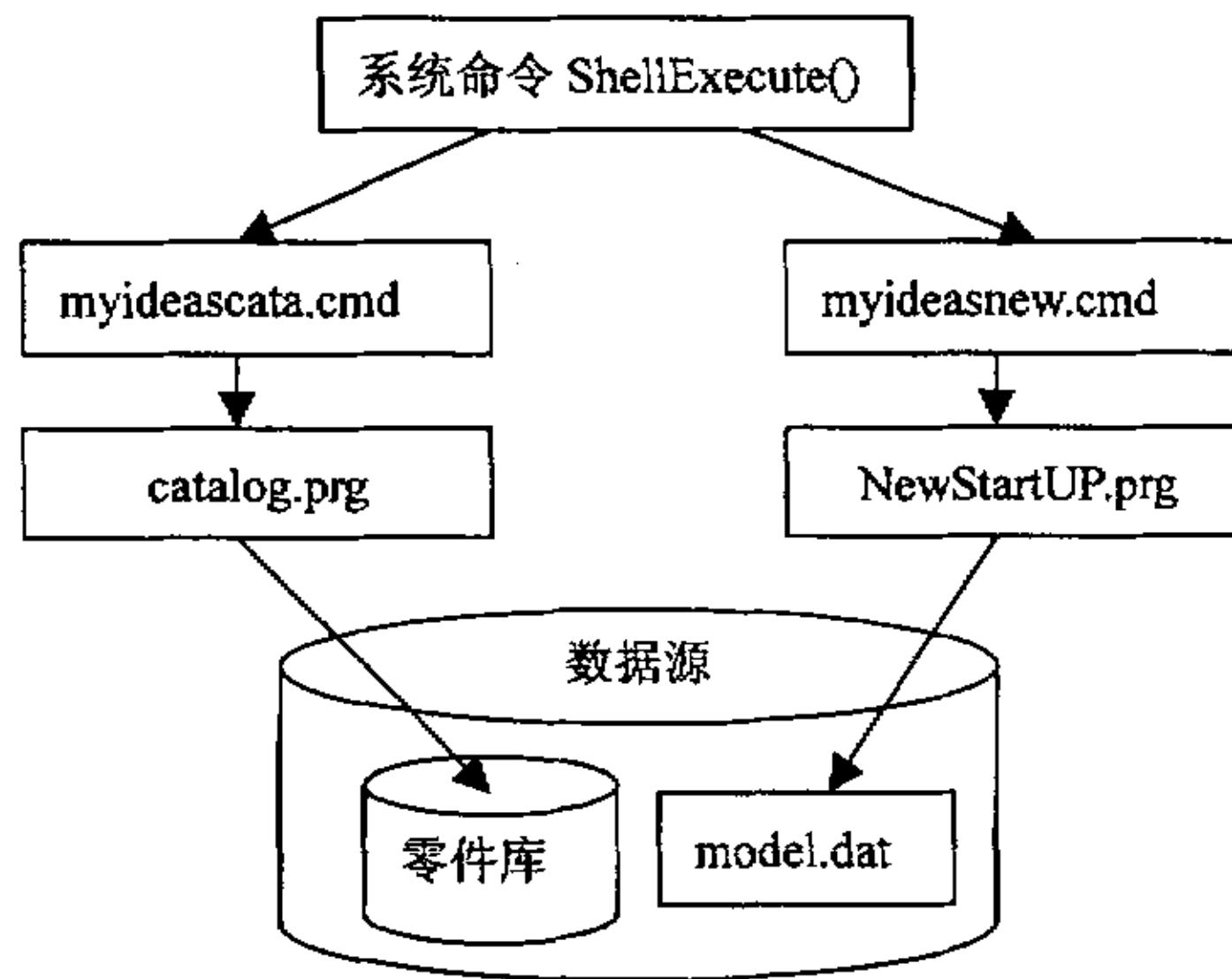


图 5.6 调用关系示意图

ideas.exe 命令行的参数说明如下^[42]:

-d 指定一个显示设备。如果在批处理模式下不指定显示设备, 缺省为 X3D, 但 X3D 不能以实体模式显示, 故这里用 OGL 设备;

-p 设定项目名;

-m 设定模型文件名;

-a 设定应用程序名;

-t 任务名;

-e 指定 I-DEAS 启动后自动执行的命令;

/f 指命令面板中的 File 菜单;

pr 指 File 菜单中的 Program File 命令;

r 指 Program File 命令下的 Run 命令;

fil 指 Run 对话框中的选择文件命令, 后面紧跟需要运行的 PRG 文件;

按照软件的界面设计原则和方法设计程序的主界面, 从而完成整个微型车驱动桥智能化设计软件系统的设计。

6 系统的运行过程及实例测试

6.1 系统运行环境

硬件运行环境为：

- (1) 微机要求主频 166MHz 以上；
- (2) 3D 图形加速卡；
- (3) 硬盘 4G 以上；
- (4) 内存 64M 以上。

软件运行环境：

- (1) 操作系统：Windows NT 4.0/ Windows2000；
- (2) 数据库：MS Access 2000；
- (3) 图形处理软件：I-DEAS 集成软件系统。

6.2 系统运行过程

- (1) 启动微型汽车驱动桥智能化设计软件系统后，主界面如图 6.1：

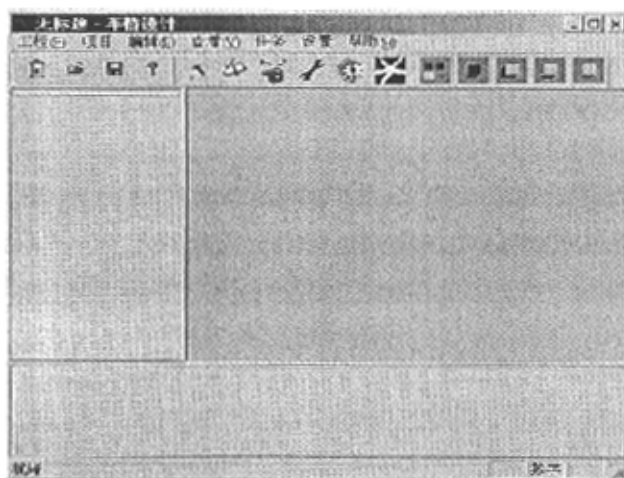


图 6.1 主界面

- (2) 执行菜单‘项目’—‘新建项目’，系统将出现如图 6.2 对话框，要求输入项目名，默认为 newproject1，并要求选择是设计驱动桥还是非驱动桥：

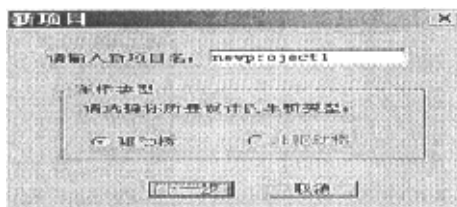


图 6.2 新建项目

(3) 若是第一次运行此系统，在执行(2)时，会在图 6.2 的对话框出现之前提示初始化，如图 6.3。若选择‘是’，开始初始化。若选择‘否’，则直接显示“新项目”对话框，但在以后选择进入三维建模时会继续提示。

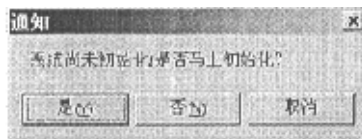
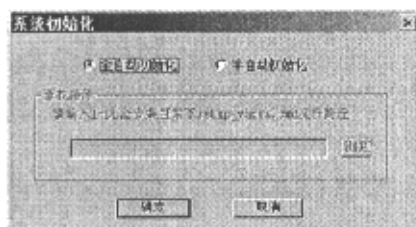
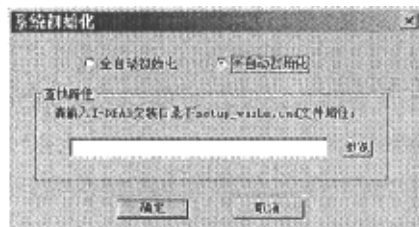


图 6.3 提示初始化

(4) 若图 6.3 中选择‘是’，弹出如图 6.4 所示对话框。图 6.4(a)是选择“全自动初始化”的对话框，系统将自动搜索 I-DEAS 系统可执行文件路径。图 6.4(b)是选择“半自动初始化”的对话框，用户将手动查找 I-DEAS 系统可执行文件路径。点击“浏览”可以浏览文件夹。单击“确定”后，系统自动启动 I-DEAS 系统，用户让其自动运行大约 20 分钟后（视主机性能决定），I-DEAS 系统会自动退出，并显示“新项目”对话框。



(a)



(b)

图 6.4 系统初始化对话框

(5) 在“新项目”对话框中选择“下一步”，则出现“选型方案”对话框，如图

6.5. 按“下一步”后，系统将记下设计者的选型方案。



图 6.5 选型方案对话框

(6) 按“下一步”后，跳出“设计方案选择”对话框，如图 6.6。可选择“从整车参数开始设计”、“直接输入关键参数”、“直接输入尺寸”；

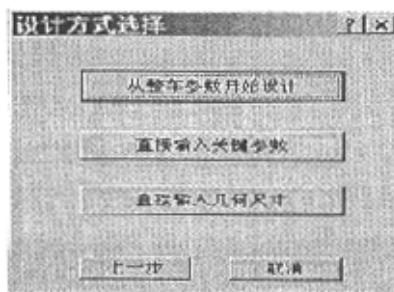


图 6.6 设计方式选择对话框

(7) 选择“从整车参数开始设计”则跳出“原始参数获取”对话框，如图 6.7。对话框共有五个属性页，分别是：主减速器设计、差速器齿轮设计、半轴设计、桥壳设计、轴承选择。带有复选框的参数在更新数据时可自动推导出一个比较合理的值，若复选框被选中，则将直接采用用户输入的值，将鼠标指在每个参数名上，停留一段时间后会现此参数的说明。在设计者输入或改动编辑框中的数据后，必须按一下“更新数据”按钮，这样相关的参数会被计算更新，计算大约需要几十秒，灰色的编辑框

在“更新数据”后可能会有所变动。按下“校核要求”，会出现校核对话框，如图 6.8。若选中“使用默认值”复选框，则系统将直接采用编辑框中给出的数据。点击“全选”，所有编辑框前的复选框都被选中，点击“不选”，所有编辑框前的复选框都不被选中。

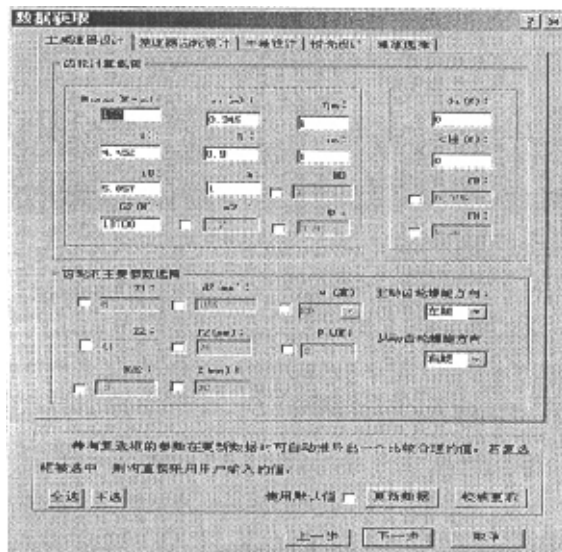


图 6.7 原始数据获取对话框

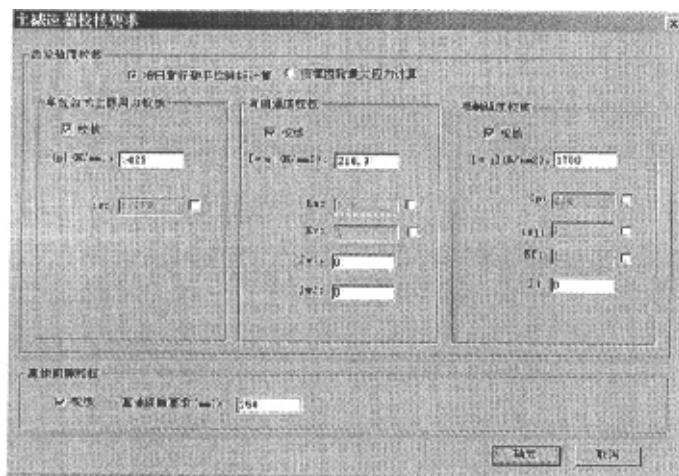


图 6.8 主减速器校核要求对话框

这是主减速器校核要求对话框，有齿轮强度校核和离地间隙校核两相。其中，齿轮强度校核包括：单位齿长上圆周力校核，弯曲强度校核，接触强度校核。齿轮强度

校核可选择“按日常行驶平均转矩计算”或“按锥齿轮最大应力计算”，选择前者和后者的许用值是不同的。若需要对某一项进行校核，则须选中“校核”前的复选框，否则运行中不进行相应的处理。

(8) 下一步后, 会给出一个“计算清单”对话框。

(9) 在“计算清单”对话框点击“下一步”后, 整个驱动桥初步设计完成, 主界面主要包括下面几部分: 菜单栏, 工具栏, 项目工作区(左), 图形显示区(右), 尺寸列表(底)。主界面显示如图 6.9, 项目工作区如图 6.10, 尺寸列表如图 6.11。

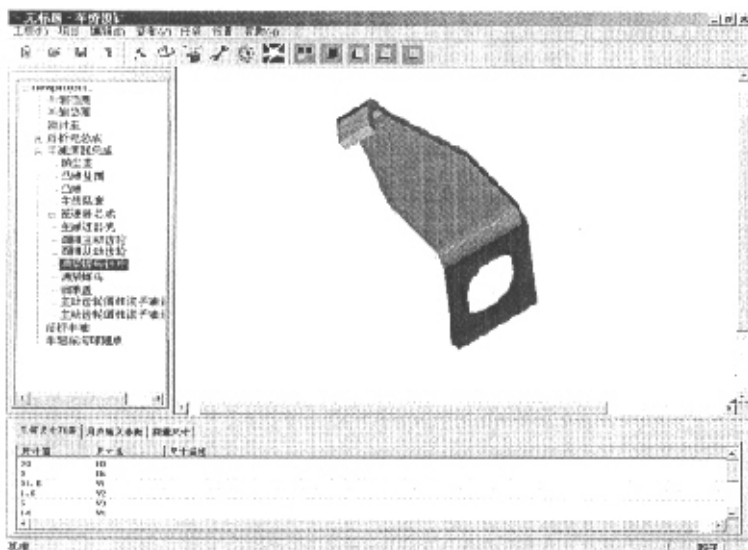


图 6.9 初步设计完成后的主界面



图 6.10 项目工作区

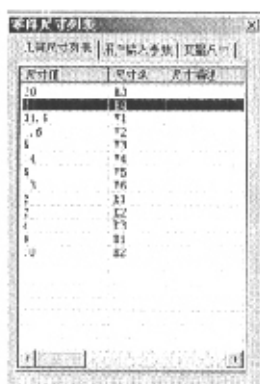


图 6.11 零件尺寸列表

(10) 在工作区中选择零件，尺寸列表中会显示此零件所有几何尺寸值，用户

可选择某个尺寸值,用鼠标单击或按空格建后可对尺寸值进行编辑。在修改完所有需要修改的参数后,必须点击“编辑”—“更新相关参数”,进行相关参数的更新计算。

(11) 在工具条上选择视图类型则主界面上将显示相应零件的视图。如图 6.9 右侧,在项目工作区中选择了“主减速器总成”中的“调紧螺母锁片”,工具栏中选择了显示立体图,显示区相应显示“调紧螺母锁片”的立体图,尺寸列表中相应显示它的所有尺寸。

(12) 输完尺寸之后,执行菜单“查看”—“生成三维实体”可进入 I-DEAS 系统自动画出零件图。此时,若在(3)中选择“否”,则会在进入 I-DEAS 系统之前跳出图 6.3,选择“是”,开始初始化。初始化完毕,系统启动 I-DEAS,启动后如图 6.12。这时用户需要选择一个 project 并新建一个 model file,Application 选择 Design,Task 选择 Master Modeler。设计的零件将在这个文件中画出。

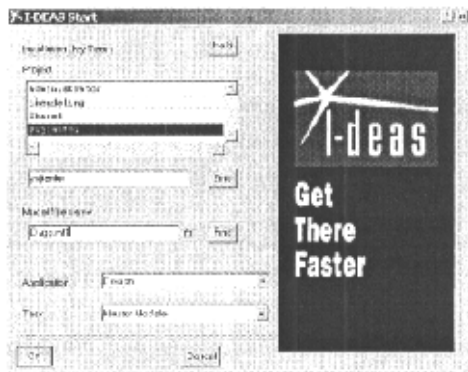


图 6.12 启动 I-DEAS 后界面

(13) 单击“OK”,I-DEAS 系统开始自动运行,大约 20 分钟后,可以输出驱动桥三维实体图,如图 6.13 和 6.14。

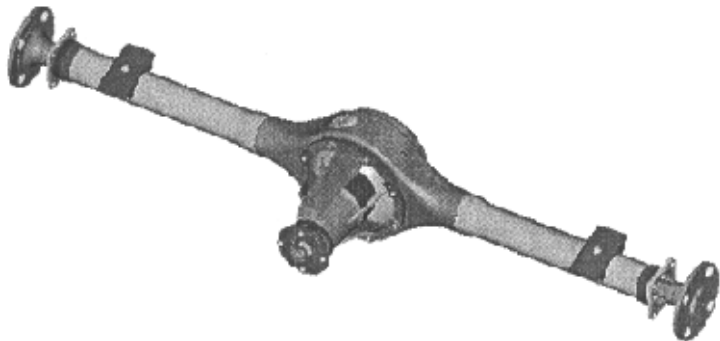


图 6.13 设计完成后的驱动桥总成

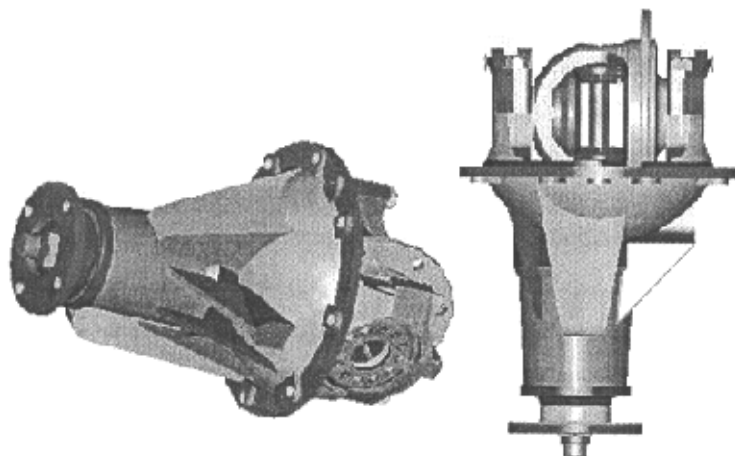


图 6.14 设计完成后的主减速器总成

由上图可以看出，本设计系统所形成的微型汽车驱动桥总体和组件结构形式合理，再加上前章有限元分析和参数约束的自动生成，本系统已达到对驱动桥进行工程化设计要求，可以为驱动桥生产企业服务。

7 总结

本文以软件工程思想为基础,详细研究了微型汽车驱动桥智能化设计系统的设计方法,提出了比较可行的设计思路。

7.1 论文的工作总结

本文首先分析了传统驱动桥设计方法的不足之处,以及对国内外相关领域的分析,说明了微型汽车驱动桥智能化设计系统研究的实用性、可行性和迫切性。

本文从概要设计、详细设计、几何建模及分析和系统的集成方法四部分详细阐述了微型汽车驱动桥智能化设计系统的设计思路。

概要设计是对微型汽车驱动桥智能化设计系统的总体把握,从分析功能需求入手,画出功能需求数据流图,然后按照数据流图划分功能模块。

详细设计给出了主要功能模块的具体实现方法。这也是本文最重要的一部分,很多重要的概念和算法都在这部分给出。选型方案的提出使得系统能够从零件库中选择正确结构的零件,并且使得零件库的扩充更为方便。数据库的设计非常重要,它是系统能运行的数据驱动源。数据库的设计尝试了面向对象的设计方法,设计的结果主要是四个表:参数表,参数关联表,零部件表,满足方案表。这四个表将各种数据以比较高的效率组织了起来,可以说数据库存放的是驱动桥设计过程中遇到的所有参数数据(包括参数值、参数名、参数间的关系等)。另外,还建立了尺寸约束函数库和知识约束库。尺寸约束函数库存放的是尺寸参数的函数关系,知识约束库存放的是非尺寸参数计算所需的专业性知识(包括经验计算公式、描述性经验值等)。

在详细设计部分还给出的数据值更新算法,是整个软件系统的核心。本文首先建立数学模型,将问题用数学语言描述出来。对数学模型进行分析并结合驱动桥设计的实际情况,设计出搜索算法和对搜索到的参数进行计算的流程。

几何建模分析部分论述了基于 I-DEAS 的参数化建模方法,并举例说明了对驱动桥的主要受力部件桥壳进行有限元分析的方法。

系统的集成方法部分主要解决与 I-DEAS 的接口的实现,这主要是采纳了前人已经提出的设计方法来实现的。

7.2 不足之处及对其的思考

本文提出的采用调用零件库中存放参数化零件来生成三维图的方式还存在不少缺陷。

由于零件库的灵活性比较差,各种零件不同的结构形式均需要建立较通用的参数化模型,零件库变得非常庞大。如果零件的每种结构形式都能够建立起比较理想的参

数化模型，那么零件库庞大一点也无妨了。问题是，有些零件的参数化模型并不是那么理想。例如，在部分尺寸发生变化时，零件模型可能会发生畸变，这样就达不到预期的效果。

对于这个问题，我们也思考过许多。例如，可以考虑不用零件库的方式，而改用在设计过程中调用绘图函数直接绘出零件图，这样就比较灵活，可以随设计过程中的具体情况而定零件的结构形式。绘图函数可以考虑使用 OpenGL 之类的图形生成工具软件库。在用这类图形生成软件库作出标准的图形后，可以通过接口，导入 I-DEAS、UG、Pro/E 等现有的大型软件系统，这样就可以随心所欲地在这些软件系统中运用建好的各类模型了。

附录 NewStartUP. PRG 源程序

```

C:-----打开零件库
K:$ /ca p
K: CA
K: CT PC
K: C Shared
K: L microaxle
K: OKAY
C:-----变量声明、初始化
K: # delete ALL
K: # declare partNumber
k: # declare index
k: # declare index1
k: # declare J
K: # declare TypeID(44)
K: # declare parasNum(44)
K: # declare paraValue(600)
C:----- 从接口文件中读取数据
K: # index=1
K: # J=1
k: # fileName="test.dat"
K: # open fp fileName
K: # IF (Z_ERROR NE 0) THEN GOTO EXIT1
K: # read fp format=" (f4.0) " Number
K: # partNumber=Number
K: # output partNumber
K: # STARTREADING:
K: # read fp format=" (f4.0) " ID
K: # TypeID(index)=ID
K: # output TypeID(index)
K: # read fp format=" (f4.0)" Number1
K: # parasNum(index)=Number1
K: # output parasNum(index)
K: # index1=1
K: # MYPARAMETERS:
K: # read fp format="(f4.0)" Value
K: # paraValue(J)=Value
K: # output paraValue(J)
K: # index1=index1+1
K: # J=J+1
K: # if (index1 LE parasNum(index)) then goto MYPARAMETERS
K: # index=index+1
    
```

```

K : # if (index LE partNumber) then goto STARTREADING
K : # close fp
C:----- 装载读入的数据
k : # output "show"
k : # index=1
k : # j=1
k : # start1:
k : # index1=1
K : $ /ca p
K : CI TYPEID(INDEX);
k : # start2:
K : PA index1;
K : VA paraValue(j)
k : # index1=index1+1
k : # j=j+1
k : # if (index1 le parasnum(index) ) then goto start2
k : okay
k : # index=index+1
k : # if (index le partnumber) then goto start1
K : $ return
K : $ mpos ;; /O B
C:----- 关闭零件库
K : $ return
K : $ AU
K : $ return
K : $ mpos ;; /do so; ; dt shh appl canc
K : $ mpos ;; /v v pe of; e 3 4 5 0; okay
    
```

致 谢

本文的完成首先要感谢我的导师陈效华先生。陈老师以其广博的知识，始终给予我以深切的关心和培养。同时，还要特别感谢四川建安工业有限公司龙思源先生，在调研期间给予我悉心的指导，并提供了大量的驱动桥设计资料。

在论文撰写过程中我还得到许多老师的指点和帮助，在此对 109 教研室的全体教师表示感谢。最后还要感谢所有给予我帮助的同学。

参考文献

- 1 张洪欣. 汽车设计. 第2版. 北京: 机械工业出版社
- 2 冯辛安. CAD/CAM 技术概论. 第一版. 机械工业出版社. 北京: 1995
- 3 童秉枢. 现代 CAD 技术. 清华大学出版社. 北京: 2000
- 4 殷国富. “机械智能 CAD 的结构模型和发展趋势”. 计算机科学. 1994. 21 (3): 53-56
- 5 蔡青, 高光焘等编著. CAD/CAM 系统的可视化、集成化、网络化. 第1版. 西安: 西北工业大学出版社. 1996
- 6 赵燕伟, 王万良 陈胜勇. 参数化 CAD 系统数据模型的研究与实践. 计算机工程与应用. 1998. 5
- 7 邵平平. “基于参数化作图的图形参数化方法”. 1995. 7 (3): 202-205
- 8 何小朝, 沈梅. 张铁昌. “基于特征的通用三维参数化零件建库技术”. 机械工业自动化. 1997, 19 (3): 17-19
- 9 汪文津. CAD/CAM 技术在汽车产业上的应用. 天津汽车. 2000 (3): 9-12
- 10 孙家广等. 计算机辅助设计技术基础. 第一版. 清华大学出版社. 北京: 1990
- 11 陈祝林. 产品设计与三维 CAD 系统. 同济大学出版社. 上海: 1998
- 12 童秉枢等. 机械 CAD 技术基础. 第一版. 北京. 清华大学出版社. 北京: 1996
- 13 教育部考试中心 罗晓沛. 三级 B 类教程. 高等教育出版社. 北京: 1999
- 14 王春森. 系统设计师教程. 第一版. 清华大学出版社. 北京: 2001
- 15 林峰, 颜永年等. “基于图形数据的图形参数化方法”. 计算机辅助设计与图形学报. 1993, 5 (3): 184-190
- 16 龚微寒. 汽车现代设计制造. 第一版. 人民交通出版社. 北京: 1995
- 17 刘润东. UML 对象设计与编程. 第一版. 北京希望电子出版社. 北京: 2001
- 18 潘双夏, 张帅, 冯培恩. 基于工程约束的产品参数化建模策略研究. 计算机辅助设计与图形学学报. 2001(9): 840-845
- 19 吴学群. I-DEAS 中模块化、参数化、建库方法及其实现. 软件开发及应用. 1997(3): 34-37
- 20 周济. CAD 基础及应用. 机械工业出版社. 北京: 1995
- 21 余卉, 徐诚. 基于 I-DEAS 的枪械概念形状快速生成系统的研究. 计算机辅助设计与制造. 2001. 8: 49-52
- 22 昌庆铃. 制动器参数化设计方法研究: [硕士学位论文]. 南京理工大学. 2002
- 23 刘震. 计算机辅助枪械外形参数化设计研究: [硕士学位论文]. 南京理工大学. 1998
- 24 Roller D. An approach to computer-aided parametric design. Computer-Aided Design. Vol23. No.5 pp385-391. 1991

- 25 B Aldefeld. Variation of geometries based on a geometric-reasoning method. Computer-Aided Design. Vol20. No.3 pp117-126
- 26 陈家瑞 主编. 汽车构造 (上下册). 第3版. 人民交通出版社
- 27 陈效华, 刘心文. 基于有限元方法的微型汽车驱动桥结构分析. 机械设计与制造. 2002. 12 (6): 655-659
- 28 龚培康. 汽车拖拉机有限元法基础. 机械工业出版社. 1995
- 29 肖刚, 李学志. 机械 CAD 原理与实践. 清华大学出版社. 北京: 1999
- 30 余志生. 汽车理论. 第2版. 北京: 机械工业出版社
- 31 林峰, 卢清萍等. “利用参数化技术建立 CAD 系统”. 中国机械工程. 1995. 6 (5): 31-33
- 32 郑四发, 连小珉, 蒋孝煜. 系列化汽车变速器设计中模型参数化的研究. 汽车工程. 2000. 22 (4): 225-229
- 33 周立. 我国汽车零部件工业发展战略选择及政策措施. 汽车工业研究. 2000. (5): 20-22
- 34 方新. I-DEAS 产品三维设计指南. 机械工业出版社. 2000
- 35 Sodhi R. Turner T U. Towards Modeling of Assembly of Product Design. CAD. 1994. 26. (2)
- 36 于铁峰、葛雁平等. “面向对象技术在建筑建模中的应用”. 计算机辅助设计与图形学报. 1998. 2
- 37 马幼鸣, 王爱俊, 陈大融. 车辆变速箱 CAD/CAE 系统的开发与研究. 制造业自动化. 2000. 22 (1): 33-36
- 38 徐向阳, 高国安. 缸盖的多模型建模及其 CAD/CAM/CAE 集成技术研究. 内燃机学报. 2000 (2): 195-198
- 39 Lee. Jae Yeol; Kim. Kwangsoo. Comput Aided Des v30 n11 sep 15 1998 Elsevier Science Ltd. Exeter Engl. p883-896
- 40 张世华, 尹德婕. “CAD 应用中的问题之一: 关于 CAD 软件二次开发的研究”. 计算机辅助设计与制造. 1997 (3): 17-20
- 41 Michael J. Youn. Mastering Visual C++6. Publishing House of Electronic Industry. 1999
- 42 Ping-Yi Chao. Yu-Chou Wang. A data exchange framework for networked CAD/CAM . Computers in industry 44(2001) p131-140