

摘 要

（产品概念设计是自顶向下的产品设计过程中的重要阶段，为了支持概念设计且使设计过程变得灵活、方便，三维 CAD 系统必须支持概念设计的表达且整个设计过程支持参数化和变量化。但是现有的三维变量化设计系统对自顶向下的设计过程支持不力且很难处理变量化技术中的三维闭环约束问题。本文针对这类问题的相关理论进行了深入地研究。）

首先，从三维变量设计系统、产品布局设计两个方面，本文在总结了国内外相关的研究成果和最新研究动态，分析了当前三维变量设计系统处理变量化技术中的三维闭环约束问题算法的基础上，结合装配的层次模型，提出了基于三维变量化系统的布局模型，尝试将工程中常见的三维闭环约束问题转化为二维闭环约束问题，并为此类问题求解提供了一个解决方案。

其次，本文依据布局模型，实现了布局设计的布局面、布局草图和布局零件对象及其关系定义，（在此基础上，结合基于历史的编码/解码原理，提出了基于编码/解码的映射机制，实现了布局设计基于历史的重构，达到了概念设计中设计意图更改而导致整个设计过程随之变更的要求，同时解决了自顶向下的零件造型问题。

最后，本文以上述研究为基础，结合“九五”国家重大科技攻关项目“基于三维的产品设计系统研究开发与产业化”，在商品化的 CAD 系统 InteSolid2.0 中，完成了布局设计部分。）

关键词： 自顶向下设计 变量设计 布局设计[✓] 三维闭环约束[✓]
编码/解码的映射机制

ABSTRACT

Conceptual design is very important stage in the process of product design. To support conceptual design and make the design process more agility and more facility, 3D CAD system must support the expression of the conceptual design and the parameterization and variation of design. But the existing systems of 3D variational design hardly support the conceptual design, and solve the problem of 3-D closed loop constraints. This paper makes deeply research on relative theory about these problems.

At first, layout model is described based on system of 3D variational design by summarizing the relative results and trends of research on system of 3D variational design and product layout, analyzing the algorithms of current system of variational design dealing with the problem of 3D Closed Loop constraints in variational technology, and combing the layer model of assembly system, which provides a way for the common problems of 3-D closed loop constraints in engineering by converting the 3-D closed loop constraints into 2-D closed loop constraints

Secondly, through the definitions and relationships of layout plane, layout sketch, layout part and the theory of coding and decoding based on history, the mapping mechanism of coding and decoding is presented, which implements modeling of the Top-down designing part and the reconstruction of layout design that the whole design process modifies automatically with the change of design ideal in conceptual design.

Finally, layout design, to which the above research and practice are applied, is implemented in commercial 3D CAD system InteSolid2.0 supported by National Key Project- R&D and commercialization of 3D Product design system.

Keywords: Top-down design variational design layout design
3D closed loop constraints mapping mechanism of coding and decoding

1 绪论

1.1 课题的提出及意义

随着 CAD 技术的不断发展及其在工业领域的推广应用, 生产方式已逐渐从传统的依靠手工绘图进行设计, 大批量生产降低成本转向用计算机进行设计, 为用户提供方便、快捷、小批量产品的模式。同时, 产品的功能、结构日趋复杂, 新产品的更新换代周期不断缩短^[1], 使得产品设计在产品整个生命周期中占据了越来越重要的地位, 从而对 CAD 系统提出了更高的要求。但是, 传统的 CAD 系统却无法^[2]满足工程设计的需要, 特别是在集成制造^[3]、并行工程^{[2][3]}和敏捷制造^[4]中表现得越发突出:

1. 大部分的商品化 CAD 系统能够提供的支持仅限制在详细设计阶段, 对于产品的功能设计、符号设计和概念设计则支持不力^{[5][6]}, 虽然近年来计算机辅助概念设计取得了很大的成功, 但是却很少同支持详细设计的 CAD 系统相结合。
2. 几乎所有的商品化 CAD 系统都可以用零件来组成装配体, 体现的是一种从下到上(bottom-up)的设计策略。而真正的产品设计过程, 往往都是先有产品的功能, 再设计完功能的装配体, 最后才是零件的详细设计, 所以说设计过程并不是严格的从下到上或从上到下(top-down), 而是这两种模式的混合形式^[7]。但目前的 CAD 系统并不能够对于这样一种模式提供支持。
3. 当今的 CAD 系统仅仅是建立在参数化设计或变量化设计之上, 而其中的几何约束求解机制早就受到 Kramer^[8]和 Thornton^{[5][6]}的批评, 特别对于四连杆等典型机构中存在的三维闭环约束问题, 在目前的 CAD 系统中解决起来速度慢、求解不稳定。
4. 对于由在装配环境下有装配关系确定的产品的设计, 是传统 CAD 系统很难完成的。

因此, 本文在开发三维变量设计系统 intesolid2.0 的基础上, 针对以上问题提出了布局设计方法, 实现了支持产品概念设计的自顶向下设计过程; 尝试将工程中常见的三维闭环约束问题转换为二维约束来求解, 从而使此类三维闭环约束求解简明而稳定; 最后用布局的方法解决了在装配环境下由装配关系确定的产品设计问题。

1.2 变量设计系统研究及发展

1.2.1 基于变量化技术的 CAD 系统

目前, 变量化技术和参数化技术还都在不断地丰富和完善, 相应地, 各大 CAD/CAM 公司纷纷推出了自己的参数化造型模块, 以跟上时代的潮流。

MDT2.0 是 AutoDesk 公司于 1997 年 12 月推出的 Windows 95 和 Windows NT 环境下的 3D 设计工具, MDT2.0 不仅包含了世界上最著名的、最完整的二维绘图工具集, 而且提供了非参数化实体造型和基于特征的参数化实体造型、基于约束的装配造型、NURBS 复杂曲面造型、实体与曲面融合以及 IGES、STEP、VDA-FS 数据交换器等一系列先进的三维设计功能及工具, 圆满地将 2D 绘图与 3D 造型技术融于一体。当然, MDT2.0 还存在一些值得改进的地方。例如不支持多个文档, 造型手段不如那些老牌的工作站系统丰富。但是 MDT2.0 的价格优势将使之成为 Pro/Engineer 或 SDRC 的重要竞争对手。

SolidEdge^{[9][10]}是 UG 公司在 Microsoft Windows 操作系统上全新的 CAD 系列软件的旗舰。SolidEdge 最大的特色是它独具创意的使用者界面, 这方面 SolidEdge 比任何其他实体模型系统表现得都要好。SolidEdge 不仅可以报告模型的定位过约束和欠约束的情形, 并使约束的修改变得十分容易, 而且它还可以通过一个被称为“Show Variability”的工具用图形显示可以变化的情况。SolidEdge 可以读入 AutoCAD 或 Microstation 所产生的实体模型进行装配设计, 并且可以在装配文档中进行零件编辑。SolidEdge 还具有极为出色的出图能力, 对模型所作的修改可以自动反应在视图上。当然, SolidEdge 也具有一些缺陷, 它不能够支持一些造型功能, 如在实体模型中加入自由曲面; 其变半径倒角能力不强, 不支持零件或装配模型与工程图的双向关联不能够产生爆炸图的工程视图。

SolidWorks 97 是 SolidWorks^[11]公司推出的 Windows 95 和 Windows NT 环境下的 3D 参数化特征造型系统, 是第一个基于微机平台的高度商品化的特征造型系统。SolidWorks 97 采用了独特的特征树管理技术, 极大地方便了实体模型的构造与修改, 并能够最大限度地保持设计者的意图。在 SolidWorks 97 中, 可以实现零件、装配体和工程图间的双向关联, 其变量化技术代表了目前商品化软件中该项技术的最高水平。值得一提的是其全 OLE 风格的二次开发接口, 使得在 SolidWorks 97 上作二次开发变得极为简单, 如使加工特征识别技术达到实用水平的 CAMWorks 的开发成功就是一例。

1.2.2 变量设计系统的求解方法

三维几何约束的求解方法主要有变量几何法和几何推理法。一般来说,对没有闭环约束的情况采用几何推理法,对有闭环约束的情况用变量几何法求解。下面分别对这两种方法作一综述:

1) 变量几何法 (variational geometry)

它是一种非线性方程组整体求解的代数方法,是 Light 和 Gossard^[12]在 1982 年首先提出的。即将几何约束满足问题转化成非线性方程组的联立求解问题。由于工程问题的复杂性,转换的非线性方程组有时非常难以求解。Lee 和 Andrews^[14]提出了将最小二乘法同牛顿-辛普森迭代法相结合来求解约束方程组,其特点在于无论约束模式是否有效,总能给出一个最小二乘意义上的近似解。Rocheleau 和 Lee^[15]于 1987 年提出了交互装配建模系统,该系统让设计者交互输入零件间的配合关系来生成装配体,并从配合关系生成联接信息以进行运动学分析和动态分析,对可旋转的欠约束零件可通过增加约束方程进行特别处理。Turner^{[13][16]}在进行有误差的零件的装配时,用一种约束优化方法将约束方程组转变为线性方程组。变量几何法的优点是:能适应很大范围的约束类型,并且循环约束情形可以用求解联立方程组得到处理。但此方法存在很大缺陷:

- 1) 方程组的求解一般都是用数值迭代法,而数值迭代法中的初值和迭代步长的选取是不容易的,容易导致迭代失败或得到一个无意义的局部最优解。
- 2) 方程组的求解没有几何意义,只能得到最后的结果,无法控制求解的进行,因此当出现无解或多解甚至错解时,不能给设计者提供有益的提示。
- 3) 约束方程组属于非线性超越方程组。当零件数为 n 时,约束方程组的方程个数为 $6(n-1)$ 。故零件数目多时,问题规模急剧增大,求解相当困难。

2) 几何推理法

几何推理法又称序列求解方法,它是通过对元件进行自由度分析来求解几何约束满足问题的。序列求解方法速度快,但只能求解开环约束情况,当约束出现封闭环时,则应该联立方程组求解。

康友树^[17]采用序列二次规划方法,通过迭代和寻优找到零件合适的位置。该方法不失为一个好方法,但优化参数的选择对问题的求解影响相当大,故很难控制约束的求解。尹文生^[18]针对封闭环的情况,以序列求解方法为基础,提出变量破环法,

将设计变量约束求解方法与几何约束求解方法结合起来,使用比较方便,有一定的实际意义。吴永明^[19]提出矢量闭环法,用其建立闭环的简化模型和矢量闭环模型,从而建立闭环的最小规模的约束方程组,用 Newton-Raphson 迭代法和最小二乘法相结合求解约束闭环。

Kramer^[20]认为几何约束的传播应具备良好的可解释性,他提出“几何约束满足问题的核心是实现约束的最大分解”。基于此,Kramer 进行了几何约束机的研究,他将几何图形抽象为无向图,顶点为几何实体称为 geom,可独立确定相对形位的顶点集合称为 macrogeoms,实体的自由度以及之间的约束度以权值的形式分别记录在顶点和弧上。约束机的求解过程可分为两步:由链式搜索和环路搜索在图上找出符合 macro 条件的几何实体集,记为 macrogeoms。从约束被改变的 macro 不断寻找上层 macro,识别其约束模式,并调用相应的方法进行约束传播,从而得到所需结果。

陈立平^{[21][22]}顺承 Kramer 的观点,基于图论、稀疏矩阵理论和自由度分析,提出几何约束传播的最大归约理论,实现了几何约束系统的最大分解,以归约树的形式表达了系统内的串、并、耦合机制,并基于此开发了 RDDS 系统。

Prudue 大学的 Hoffman^{[23][24]}首先对基于几何约束的图形进行分析,生成一个约束序列,而后按此序列进行传播构造图形。他采用跟踪调试的方法,使图形生成的每一步均能被控制,用户可根据需要在适当的步骤中调整图形,从而得到正确的结果,但操作起来过于复杂。

1.3 布局研究现状

1.3.1 布局研究现状

从目前的研究内容来看,布局问题主要集中在二维^{[25][26]}、三维的实体布局^{[27][28][29]}(Packing Problem)、空间布局^{[30][31][32]}(Spatial Layout)和智能布局^{[33][34][35][36]}(Intelligent Layout)上,其研究成果已广泛地应用于建筑、智能布局、线路布置及优化设计等各个领域。在产品设计领域,随着现代产品设计技术的发展与消费心理的变化,一个产品布局设计的好坏,在很大程度上影响着整个产品的美观性、宜人性,因而产品布局设计的研究愈显重要和迫切。

然而,现有的布局理论并不能很好地应用于产品布局设计。原因在于:

- 1) 由于布局问题固有的复杂性, 现有的布局理论往往局限于平面布局, 而产品布局不仅是一个立体布局问题, 而且其布局空间是一个非等值空间。
- 2) 现有的布局理论一般都聚焦在抽象层次, 其布局对象是一个泛化的概念。而产品布局包含了应用层次上的问题, 必须结合具体的布局对象一起加以考虑。
- 3) 现有布局理论中的布局对象, 在整个布局过程中, 其自身属性往往是静态的。而产品布局中各个对象却是动态的, 它随着布局状态的改变而作自身调整。
- 4) 在布局设计中, 布局状态的变动与布局对象的属性变化是紧密相关、互为制约的。因而可把产品布局设计视为一个产品对象的布局演变过程。从这一观点看, 产品布局是一个四维空间的布局问题。

另一方面, 产品布局设计是设计工程技术 (Design Engineering Technique, DET) 的组成部分。在传统的机械工程设计中, 产品布局一般是在结构设计中才加以考虑甚至完全忽略。因为在机械产品中, 各个零部件大都处于一种啮合关系, 在整个产品中的布局变化能力很小。

同时, 从抽象层次看, 产品布局设计作为产品级层次的设计, 考虑的布局特征是构成产品的各个部件, 而非单个零件, 布局约束也针对产品的各个部件。然而现有的产品模型几乎都是基于零件层次。虽然两者具有不少相似属性, 但在实现方式和应用水平等方面却有着本质区别。因而要有效地解决这一问题, 关键在于将布局特征的内涵和范围从零件级拓宽到产品级, 从设计后期延伸到整个设计过程。基于此, 孙守迁等^[37]提出了基于产品级的组件特征的布局模型, 该模型将布局设计过程看作一个组件特征的布局演化过程, 将产品的总体特征划分为各种性质的布局特征来进行配合从而实现产品功能设计, 但是在初始对象的获取、布局优化设计、人机结合方面做得不好, 而且没有将此方法用来指导下层的零件详细设计。杨勤^[38]利用同一产品的各种合理的布局的特性, 建立同一产品的布局库, 从而为产品的不同特性的设计提供参考, 这样每一次设计都需要匹配不同特性的产品布局, 虽然对单个产品可以实现自顶向下的设计, 但是不适用与所有的产品设计。

1.3.2 支持概念设计的布局研究现状

当前一些大型 CAD / CAM / CAE 商品化软件系统如 EDS Unigraphics、Pro/Engineer 等也提供了概念设计 / 工业设计模块, 其设计结果能直接传递给系统中其他模块进行后续设计。

1) Pro/Engineer

当今国际上最流行的 CAD 系统, 是 PTC 公司 PRO-ENGINEER, 本系统较好的适应 CIMS 的要求, 它的另一优点是二维、三维变量设计一致贯通。确切的说, 所有的变量设计都是在三维基础上进行的, 二维的变量设计也是以三维内部运算为依据, 根据需要截取三维立体的正面、侧面或水平投影得到的。严格说来, 三维和二维内部运算分属不同的范畴, 因此 PRO-ENGINEER 的二维变量设计功能受到一定的限制。

Pro/Engineer^[39]是通过 Layout (布局草图) — Assembly Modeling (装配建模) — Part Modeling (零件造型) 顺序设计实现从上到下和功能设计的。Layout 是二维草图。他能使用户定义一个装配体的基本功能要求和全局参数及参数之间的顺序依赖关系, 以一种概念的形式声明、定义、记录了零件和装配体的简单信息, 而无须进行繁琐细节的几何设计。其零件或部件可以是一个概念上的方块图形或参数草图, 以此来代替真实的实体模型, 然后建立参数、尺寸之间的关系和零、部件自动定位所需的全局基准。例如有两个零件引用了同一个参考轴时, Pro/Engineer 就执行这两个零件的共轴配合 (Insert) 操作, 当两个零件同时引用一个基准面时, 它就执行这两个零件的面配合 (mate) 操作。这种 Layout 中建立的引用关系使装配和详细零件设计能保持最开始的、最基本的设计意图。

然后, Pro/Engineer 在 Layout 指导下进行零件造型^[40]和零件装配, 并在这个过程中, 设计者可以定义一些尺寸之间的关系; 进行尺寸驱动。另外 Pro/Engineer 还有一个置换功能, 他可以用一个复杂零件或部件替换原始二维或三维草图勾勒出的零、部件, 那么, 可以看出 Pro/Engineer 的功能设计和三维实体造型之间只有全局参考和基准的联系面而无几何实体的对应关系。因此, 当进行三维实体设计时, 设计者必须时刻调出全局参数的关系。这样, Layout 实质上只是一种传统图纸设计中的粗略总装图。再考察零件设计和装配中的尺寸关系及尺寸驱动。Pro/Engineer 有四个限制要求: 1 如果关系驱动装配体中的零件, 则此关系只能在装配环境中有效; 2 不能用装配关系产生零件实例的驱动尺寸; 3 在零件中已被零件关系驱动的参数不能在装配中再产生装配关系驱动; 4 如果零件内加入了一个尺寸驱动关系, 而此关系又已在装配体中被关系驱动, 则装配级关系驱动无效。另外, 这些关系只能以 $d_i = f(d_1, d_2, \dots, d_j, \dots, d_n), i \neq j$ 的形式出现, 并且在求解 d_i 时认为 $d_1, d_2, \dots, d_j, \dots, d_n$ 为已知, 这说明这些尺寸的关系是

顺序求解。由此可看出, Pro/Engineer 是依赖用户输入的关系的顺序来求解装配体和零件中的尺寸, 而作为一个面向功能的设计, 最上层输入的功能约束(一般以 $F(d_1, d_2, \dots, d_i, \dots, d_n) = 0$ 形式出现) 无论在任何时候都应被满足而不应被后续可能输入的 $d_1, d_2, \dots, d_i, \dots, d_n$ 中任一尺寸的依赖关系破坏, 况且在 Pro/Engineer 的装配建模中上述功能约束形式是无法输入的。

由此可见, Pro/Engineer 在支持产品功能设计上是无法用功能参数来完全有效地实现的, 而它的功能置换实际上也只是一种几何形体的置换, 体现一种从上到下的思想。所以, 它以草图、功能形状置换有效地解决了形体的从上到下的设计过程, 但功能的从上到下及子功能间的联系无法完善地体现。

2) Solidworks 三维造型系统

SolidWorks 公司推出 Windows 环境下的 3D 参数化特征造型系统 Solidworks 2001, 是从第一个基于微机平台的高度商品化的特征造型系统 Solidworks 97 不断升级而得到的。SolidWorks 2001 采用了独特的特征树管理技术, 极大地方便了实体模型的构造与修改, 并能够最大限度地保持设计者的意图。在 SolidWorks 2001 中, 可以实现零件、装配体和工程图间的双向关联, 其变量化技术代表了目前商品化软件中该项技术的最高水平。它同样支持概念设计, 并将概念设计用来指导产品的详细设计。

在 Solidworks^[11] 系统中, 遵循 Assembly—Layout—Part 的设计过程来实现自顶向下的产品设计。任何产品都是有具有某些功能的装配体, 功能对产品要求可以认为是功能对一装配体的要求, 所有满足功能的产品设计都应该是满足功能的装配体设计, 因此自顶向下的设计实际就是自顶向下的装配体设计。用布局表示组成功能的组件和它们之间的配合以及距离关系, 给出了粗略的概念设计, 用布局来表达概念设计对零件详细设计的指导设计。因此必须存储布局信息, 在存储布局信息时将它完全映射到零件环境中为零件设计提供设计参考。但是零件环境中的零件设计只能参考布局草图信息, 因此零件设计必须要重新绘制零件草图, 布局对零件草图的控制是建立零件草图与布局草图的一种全等约束, 这种全等约束对于简单语义的图形元素(比如: 圆、点等)来说比较简单, 但是对于复杂语义(如: 矩形)构造起来相当复杂, 有时甚至不可能。并且在零件设计时无法用布局草图的元素, 这样的话布局体现的只是概念设计的位置约束, 而没有表达产品的零件信息, 而真正的概念设计则是零件的简单形状和他们约束关系的表达, 可以说他不是完全意义上的概念设计。

3) EDSUnigraphics

它从 V13 版本后推出了概念设计 (What-if Alternative Value Engineering, WAVE) 技术。WAVE^[41]为协同概念设计提供了强大的技术支持。它使不同部门的工程师在概念设计阶段,从系统工程的角度同时对多种可选的设计方案进行评估。通过将设计意图组织到一个相关控制结构中去,可使概念设计师有效地控制各种设计的变更。此外,WAVE 也支持“概念设计到详细设计工程”;即先作出设计决策,然后设计细节。

1.4 本文的工作

本文的工作是结合国家“九五”重大科技攻关项目“三维变量化计算机辅助设计系统”“天喻三维 CAD”项目,对华中科技大学 CAD 中心和天喻软件公司商品化三维软件天喻三维 CAD 系统的布局设计进行研究和开发。此开发的目的是为了在系统中实现支持概念设计的产品设计,并使工程中常见的三维闭环约束问题的求解变得简捷而有效。

本文主要完成以下工作:

- 1) 总结国内外的三维变量化设计 CAD 系统以及变量求解方法,分析它们对三维闭环问题的求解不力,提出了支持概念设计的布局求解方法,将工程中常见的三维闭环问题化解为二维闭环约束问题进行求解,从而使此类闭环问题的求解变得简明而有效。
- 2) 研究了基于历史的布局对象定义,布局面、布局草图、布局零件之间存在约束与被约束的历史关系。有了这些关系才可以实现概念设计对零件的详细设计控制。
- 3) 在研究了布局设计的编码/解码原理的基础上,提出了基于历史的编码/解码映射机制,此机制维护不同设计环境的编码/解码的一致性,从而实现装配环境布局的变更快速的反应到零件环境产品零件的变更。

1.5 本文的组织与结构

第一章: 本文从三维变量设计系统、产品布局设计两个方面总结了国内外相关的研究成果和最新研究动态,分析当前国内外的三维 CAD 系统存在的问题,阐述本课题的提出和意义。

第二章：分析目前三维变量设计系统采用的约束求解方法对工程中常见的三维闭环约束问题求解不力，提出了求解此类闭环约束的布局方法，并给出布局的数学描述以及基于层次装配结构的模型。

第三章：本文依据布局模型，实现了布局设计的布局面、布局草图和布局零件对象及其关系定义。

第四章：本文采用基于历史的编码/解码原理对布局元素的历史建立过程进行记录，在此基础上提出了基于历史的编码/解码映射机制，实现了装配环境与零件环境的编码一致性，从而为维护布局设计历史过程提供理论依据。最后根据此原理和机制，建立了布局历史过程维护算法并给出了实例，结果表明布局设计的历史过程得到正确的维护。

第五章：简要介绍三维变量设计系统及其布局各对象的功能和实现步骤，并举例说明如何用布局来实现对产品概要设计的支持。

第六章：总结全文，阐述本文的研究成果并展望进一步的工作。

2 基于变量设计的布局模型

【本章摘要】本章针对目前三维变量系统很难求解闭环约束问题，提出了布局的求解方法。结合布局的数学描述，构造了基于开放式变量设计装配模型的布局设计框架。最后用实例论证这一模型能简明而快速的解决工程中常见的三维闭环约束问题。

2.1 引言

装配设计的关键是得到装配体内各个零件的正确位置，而得到零件正确位置的工作就是通过当前的空间位置计算得到坐标变换矩阵，把零件从其局部坐标系下变换到装配体的全局坐标系下，并且满足在装配设计中定义在这些零件之间的配合约束关系。三维变量系统求解的目标就是求出符合条件的各零件的位置变换矩阵，三维变量系统的求解的结果直接决定了装配结果的好坏。

2.2 三维变量系统求解

2.2.1 基于自由度的几何推理方法

文献^[17]提出的基于自由度的三维几何约束推理和三维几何约束序列二次规划求解两步求解方法。首先把装配体内的三维几何约束转化成带连通块结点的多叉树，这样就在整体上把装配图的潜在有向性用一个具有明显顺序性质的多叉树替代，多叉树的根结点到其它任意一个叶子结点的路径都是一条开链，开链约束可顺序推理求解。在开链约束中，求解任一结点零件时，他的所有上级结点都可视为固定的参考零件，这种已知上级结点位置及上级结点和子结点之间的约束，推理出子结点位置的方法就是约束推理法。

在整体上的约束图用约束推理法求解的同时，对连通块结点采用最小完全约束化处理 and 最小规模的串连接立求解，对于非完全约束集，由于其解是多解情况，我们用序列二次规划法给出非完全约束集的一个最优解。三维几何约束求解的总体流程如图 2.1 所示。

对于经过规整的约束树，它是一个从树根到树叶的有向约束树，根结点就是整个装配体的基，也可以是世界坐标系，其它树叉根结点可以看成是虚拟基，它们相对其上级（父级）虚拟基定位，而叶子结点又相对于虚拟基定位，这样就可以确定从树根到树叶的推理方向。三维几何约束推理的目的就是找出这样一个变换矩阵，它使被装

配零件从局部坐标系下变换到装配体世界坐标系下,并且这个位置满足加在被约束零件上的所有几何约束,而在推理时视所有与之相关联的零件为固定,加在一个零件上的约束有多个,个数的多少要靠约束的自由度数是否超过最大限度6来确定,而约束形式又有多种,约束在零件上的约束形式可以任意组合,针对每种约束形式各编一个矩阵求解函数,然后把实际的组合形式与所有这些预想的组合形式进行匹配,为匹配出的函数输入约束组合形式,就可得相应的矩阵。

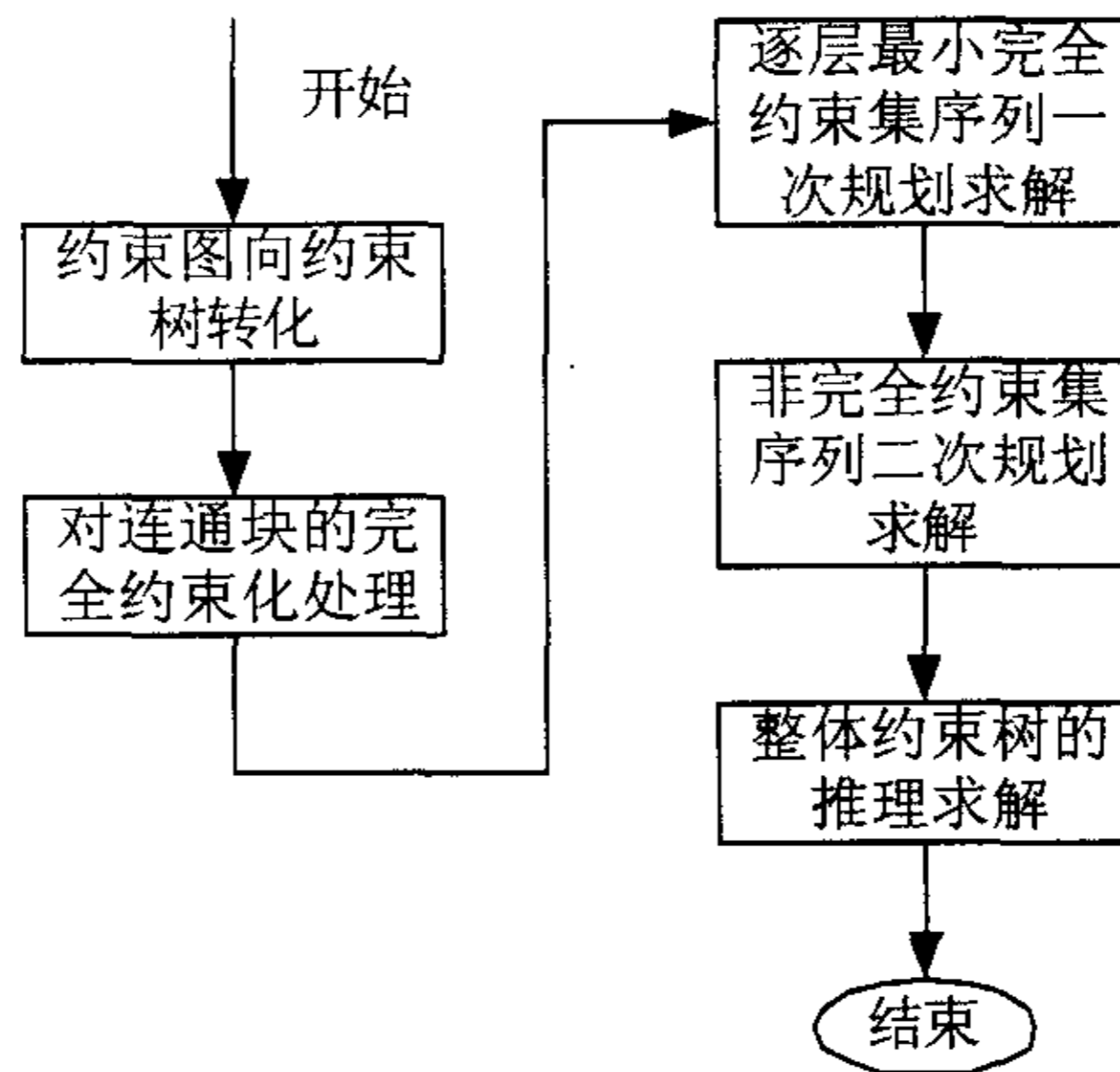


图 2.1 三维几何约束求解流程

此文献的解决办法是,一个一个单独处理加在零件上的约束,在处理当前约束时,根本不考虑前面已处理过的约束和后面待处理的约束,而只关心当前步的约束(称为单步约束)和零件所剩自由度,通过求解满足当前单步约束矩阵来求解最终矩阵。约束推理流程如图 2.2 所示。

基于自由度的三维几何推理最擅长的是对开链进行推理求解,但是对于连通块,其中存在一条或多条闭链,虽然在联立求解之前可以从闭链的某一环断开,使其成为开链,从此断开链开始依次向下推理求解,这种推理不能求得最终的精确解,只可以为联立求解提供一个非常优势的迭代初始值,因此几何推理法无法解决闭环问题。

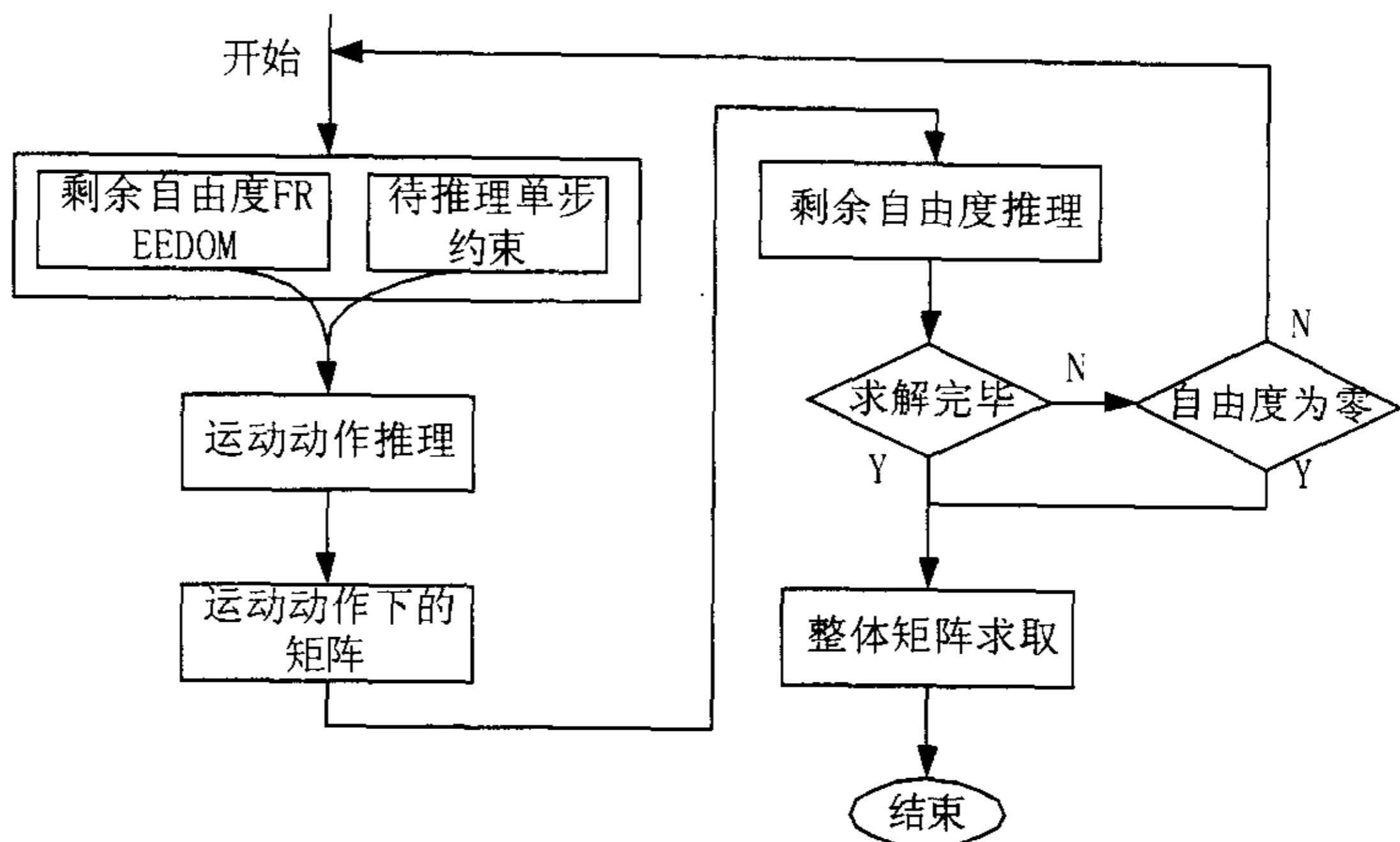


图 2.2 约束树推理求解

2.2.2 变量几何方法

变量几何法是一种面向非线性方程组整体求解的代数方法, Hillyard^[42]提出了约束的非线性方程组表达, 约束系统可以表达为状态变量 $X = \{x_j | j=1, 2, \dots, n\}$, 和一组约束方程 $F = \{f_i | i=1, 2, \dots, m\}$, 其中 $m \leq n$ 。即约束系统可以表示为:

$$F(X) = 0 \quad (1.1)$$

其中: $X = (x_i)_{i \leq n}$, $F = (f_i)_{i \leq m}$

由 MIT 的 Gossard^[43]研究小组发展和完善了这一理论, 称为变量几何法 (Variational Geometry)。该方法把几何形状定义成一系列的特征点, 约束则表示成以特征点坐标为变元的非线性代数方程组, 通过数值迭代方法求解非线性方程组, 从而确定出几何细节。

非线性方程组的求解通常基于 Newton-Raphson 迭代法。对于式(1.1)的方程组, 由初值 X^0 , 第 $n+1$ 次迭代的 X^{n+1} 值由以下迭代公式得到:

$$X^{n+1} = X^n - [F'(X^n)]^{-1} F(X^n) \quad (1.2)$$

其中, $F(X^n) = (\frac{\partial f_i}{\partial x_i}(X^n))_{i=1 \dots m}$

令: $\Delta X = X^{n+1} - X^n$

则有:

$$\begin{cases} F(X^n)\Delta X = -F(X^n) \\ X^{n+1} = \Delta X + X^n \end{cases}$$

我们于是得到:

$$J\Delta X = f \tag{1.3}$$

其中, $J = \begin{bmatrix} \frac{\partial F_1}{\partial x_1} & \frac{\partial F_1}{\partial x_2} & \dots & \frac{\partial F_1}{\partial x_m} \\ \frac{\partial F_2}{\partial x_1} & \dots & \dots & \dots \\ \vdots & & & \\ \frac{\partial F_m}{\partial x_1} & \dots & \dots & \frac{\partial F_m}{\partial x_m} \end{bmatrix}$ 为雅可比矩阵。

$\Delta X = \{\Delta x_1, \Delta x_2, \dots, \Delta x_m\}^T$, 表示各个变量的微小位移。

$f = \{-F_1, -F_2, \dots, -F_n\}$, 表示方程组的残差。

经过反复迭代, 直至 $|\Delta X| \leq \varepsilon$, 就得到了方程组的解。约束集的有效性可以通过判断雅可比矩阵的秩来实现。

此方法的解决策略是, 约束系统的状态变量可以用一组变换矩阵 $M_1, M_2, \dots, M_i, \dots, M_N$ 表示, 此矩阵应该满足加在这些零件上的几何约束。设此状态下加在第 i 个零件上的约束有 Q_i 个, 一个约束对两个元素同时产生作用, $element_{i(1)}, element_{i(2)}, \dots, element_{i(Q_i)}$ 表示被约束的零件 i 上的元素, $element_{j(1)}, element_{j(2)}, \dots, element_{j(Q_i)}$ 表示与被约束元素相对应的其它零件上的元素, $DIM_{i(1)}, DIM_{i(2)}, \dots, DIM_{i(Q_i)}$ 表示约束尺寸, 定量约束的约束尺寸为约束的量,

定性约束的约束尺寸为 0。因此约束系统中零件 i 的状态方程为：

$$\sum_{k=1}^{Q_i} \left| \left(element_{i_1}(k) \times M_{i_1} - element_{i_j}(k) \times M_{i_j} \right) - DIM_i(k) \right| = 0 \quad (1.4)$$

其中 $\left| element_{i_1}(k) \times M_{i_1} - element_{i_j}(k) \times M_{i_j} \right|$ 表示 $element_{i_1}(k)$ 与 $element_{i_j}(k)$ 在位置 M_{i_1} , M_{i_j} 上的距离值或角度值。距离与约束尺寸差的绝对值 $\left| \left(element_{i_1}(k) \times M_{i_1} - element_{i_j}(k) \times M_{i_j} \right) - DIM_i(k) \right|$ 就是零件约束违反量，因此有 N 个零件的约束系统的状态方程为：

$$\left\{ \begin{array}{l} \sum_{k=1}^{Q_1} \left| \left(element_{1_1}(k) \times M_{1_1} - element_{1_j}(k) \times M_{1_j} \right) - DIM_1(k) \right| = 0 \\ \sum_{k=1}^{Q_2} \left| \left(element_{2_1}(k) \times M_{2_1} - element_{2_j}(k) \times M_{2_j} \right) - DIM_2(k) \right| = 0 \\ \dots\dots\dots \\ \sum_{k=1}^{Q_i} \left| \left(element_{i_1}(k) \times M_{i_1} - element_{i_j}(k) \times M_{i_j} \right) - DIM_i(k) \right| = 0 \\ \dots\dots\dots \\ \sum_{k=1}^{Q_N} \left| \left(element_{N_1}(k) \times M_{N_1} - element_{N_j}(k) \times M_{N_j} \right) - DIM_N(k) \right| = 0 \end{array} \right. \quad (1.5)$$

给系统输入一组初始状态矩阵 $M_{1_1}, M_{1_2}, \dots, M_{1_j}, \dots, M_{1_N}$ ，用此初始值代入公式 $X^{n+1} = X^n - [F(X^n)]^{-1} F(X^n)$ 进行迭代，以此求解满足三维变量设计系统约束的零件最终位置状态矩阵。

由上算法过程可知，如果当 $m > n$ 时，方程组有无穷解，因此它只能求解完备约束问题，不能求解欠约束问题，这样使得变量几何方法很难应用于实际的 CAD 系统中。但是即使对于完备约束如三维闭环问题而言，一方面，用户很难将公式(1.4)具体化来刻画几何元素之间的约束关系，并且当出现不一致的情况或非预期的结果时，从约束方程组得不到准确的反馈；另一方面，由于迭代初值与步长的选取也会影响算法的成败，因此数值方法求解稳定性差，方程组整体求解的规模和速度难以得到有效的控制。

由上述两种三维变量系统求解方法可知，闭环约束问题一直是三维变量设计系统中的一个难解。针对目前系统对闭环约束问题求解的困难，本文提出布局的方法，该

方法是基于几何推理的求解器的基础上,将三维几何约束转化为二维几何约束进行求解,从而将空间的多自由度约束满足问题转化为平面自由度约束满足问题,这样简化了闭环约束问题的求解,从而可为工程中常见的三维闭环约束问题提供求解方案。

2.3 布局模型

2.3.1 布局的数学描述

定义 2.1: $A = (L, G)$ 为产品布局设计;

定义 2.2: $L = \{LS_1, LS_2, \dots, LS_n\}$ 为 N 个布局草图, LS_i 表示第 i 个布局草图;

定义 2.3: $G = \{(LS_i, LS_j), LS_i, LS_j \in L, i, j \in n, i \neq j\}$ 为布局草图之间的约束关系,

其中 (LS_i, LS_j) 表示两布局草图之间的尺寸等约束;

定义 2.4: $LS = \{V, R(E)\}$, 其中 $V = \{E_1, E_2, \dots, E_i, \dots, E_m\}$ 为 M 个布局, E_i 表示第 i 个布局; $R(E) = \{(E_i, E_j); E_i, E_j \in V, i, j \in m, i \neq j\}$ 为布局之间的约束关系; (E_i, E_j) 为一个布局容器中第 i 个和第 j 个布局之间的约束关系;

定义 2.5: $B = \{BS_1, BS_2, \dots, BS_k\}$ 为 M 个布局映射到零件环境下对应 K 个基草图;

定义 2.6: $F = \{F_1, F_2, \dots, F_l\}$ 基草图所形成的基特征。

由以上定义可知,产品设计可以认为是布局设计,一个产品可以根据它功能的划分而用多个布局进行表达,布局是用二维草图来表达产品零件和零件之间的约束关系,从而指导三维产品设计。其中 LS 只能在参考平面上生成, (LS_i, LS_j) 通过布局草图之间的尺寸约束等来实现; 一个布局草图元素可以表示一个布局 E_i , (E_i, E_j) 表示布局之间的约束,记录了装配体各零件的配合关系; 每一个布局 E 可以映射一个或者多个零件草图 BS ; 每一个零件草图 BS 拉伸成一个基特征 F 。就这样布局草图通过布局设计来向零件环境映射,控制拉伸零件基特征草图的生成,达到维护布局草图与零件草图信息的一致性。

任何一个产品都可以看作是一个装配体，产品的设计就是装配体的设计，装配体由零、部件按照一定的配合关系来表达，而布局设计正体现这一设计过程，产品 A 根据功能分解用 N 个 LS 进行表达， LS_i 和 LS_j 之间的约束关系 (LS_i, LS_j) 记录功能映射为关联布局草图的信息，而每一个 LS 又表达一个或多个布局 E_i 和布局间的约束 (E_i, E_j) ， E_i 映射到零件环境为一个或多个基草图 BS ，从而在零件级进行装配体的详细设计。

2.3.2 支持布局设计的开放式装配模型

布局设计是面向产品功能的设计，能够为设计者的设计过程提供强有力的支持，是在三维设计系统中体现用户的设计意图^[44]的必备条件。装配设计是三维产品设计的重要阶段。本文从以下几点来考虑支持布局设计的变量化装配模型：

1) 支持从上到下的产品形状和功能的层次划分

设计者先有产品的整体形状和功能^{[45][46]}构想，之后把这个整体形状一层一层地划分为子空间，每层子空间就代表一级子装配体，一直划分到设计者认为不可再分的子空间，这个最后的子空间就是布局。在这一层一层子空间逐级划分的同时，设计者可以把产品的整体功能一级一级向各子空间分解，形成子功能，当分解到不可再细分时的子功能就是布局零件将要在产品中起的作用。装配模型应该体现这种产品形状和功能的层次划分。

2) 支持抽象设计

设计者在刚开始设计时，得到的东西可能是非常抽象的或不完全^[50]的，详细设计可在后期加入。

3) 支持详细设计及回溯过程

设计者在产品功能设计开始后就可进行元件的详细设计。元件的详细设计^[44]能返回到原有的功能设计并可能对其作一定的修正。这就是支持功能的维护。

4) 能够捕捉设计者的设计意图

设计者对产品的设计总是从产品的功能需求开始的。然后将功能进行分解，并逐步用几何和数据加以表示。系统应该能够记录设计者的设计意图并作为以后设计的依据。

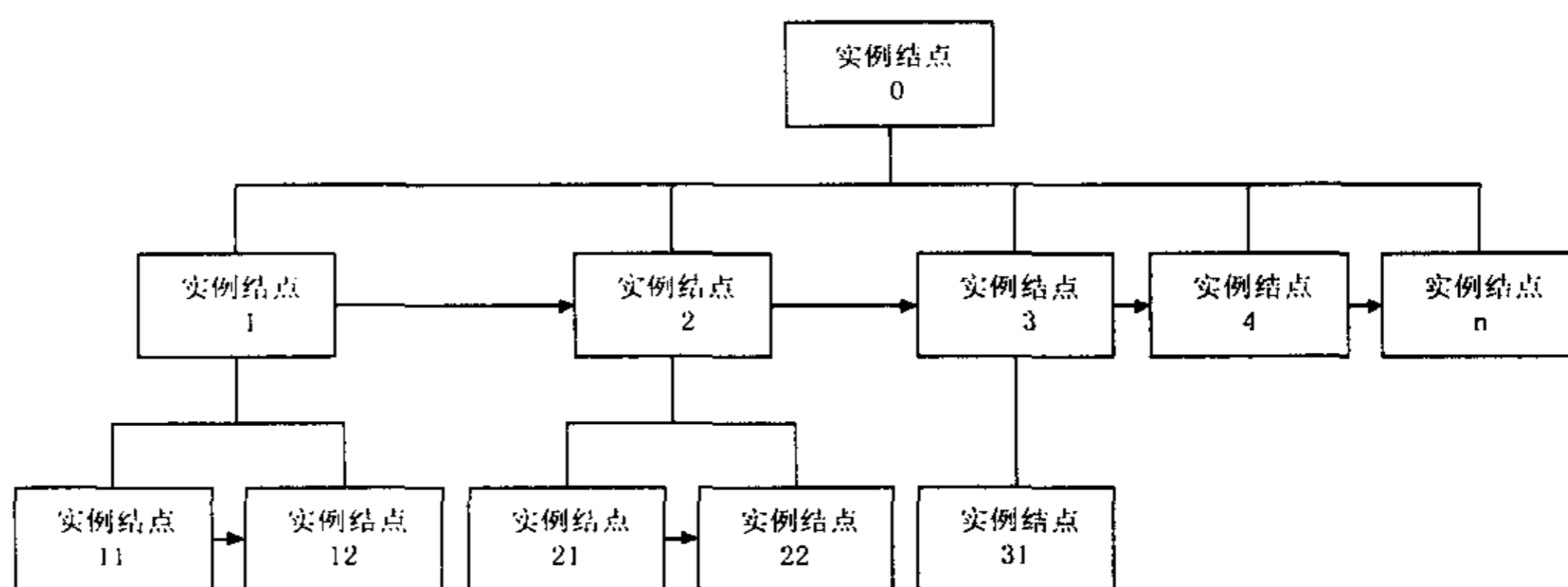
5) 开放性要好

一个 CAD 系统应有好的开放性，这样的系统才方便与其它具体应用模块（例如标准件库）集成，从而实现不同的实际应用。CAD 系统的开放性取决于其系统模型。

要提高系统的开放性，就应以开放性^[47]为目标，建立开放式的系统模型。

建立合适的装配设计模型是实现布局设计的基础。已有的装配模型一般来说分为图模型、层次模型和混合模型三大类，其中以 Lee 和 Gossard^[49]提出的混合模型虚链结构最为著名。图模型充分反映了约束关系，但缺乏层次信息，管理比较困难；层次模型则相反，不能记录复杂的约束关系；以虚链结构为代表的混合模型充分结合上述两种模型的优点，但在处理复杂的布局设计时仍显得力不从心。

尹文生在其博士学位论文^{[18][48]}中提出的基于广义环图树的装配模型是一种由抽象元件结点构成的层次树装配设计模型，它是三维造型系统 InteSolid2.0 的基础装配模型。本文在这一装配模型的基础上，在抽象元件结点中增加布局及其相关设计对象的抽象结点，以记录和处理产品设计过程中的布局设计和相关的布局面、布局草图的设计。一方面充分利用其自顶向下的分层特点，另一方面考虑到模型的开放性，运用组件思想，引入利于布局设计的开放式装配模型。如图 2.3 是基于广义环图树的装配模型，保留它自顶向下的分层特点，改变其抽象结点的具体结构和实现方法后得到图 2.4 开放式装配模型总体结构。



现具体讲解一下开放式装配模型：图中每个方框中单词的第一个字母 H 表示类名前缀，无特别意义，HL*表示各类组件的代理，相应的对象类从 HASLINK（代理基类）派生，HC*表示各类组件与系统的接口，相应的对象类从 HCOMPONENT（各种组件的基类）派生。图中不带箭头的表示下层继承上层，带箭头的表示引用关系。HINSTANCE 代表实例结点。整个装配模型的组件分为两类，核心组件和扩展应用组件。核心组件是基本组件，是系统运行所不可缺少的部分，包括有：零件组件、装配体组件、草图组件、零件布局组件、配合组件、变量设计组件等。扩展应用组件可根据需要向系统挂接。在装配设计中，装配序列规划、机构分析与仿真、运动仿真等功

能都是十分有用的，可以通过挂接扩展应用组件来得到。每个组件通过各自的代理与系统相连接。

总的来说，开放式装配模型有如下优点：

- 1) 数据结构简单，清晰地表达了装配设计过程的层次化结构。
- 2) 降低了系统开发难度，使系统变得更易于维护。
- 3) 软件可重用性提高，从而使系统应用效率提高。
- 4) 增加了系统的灵活性，使系统易于集成。

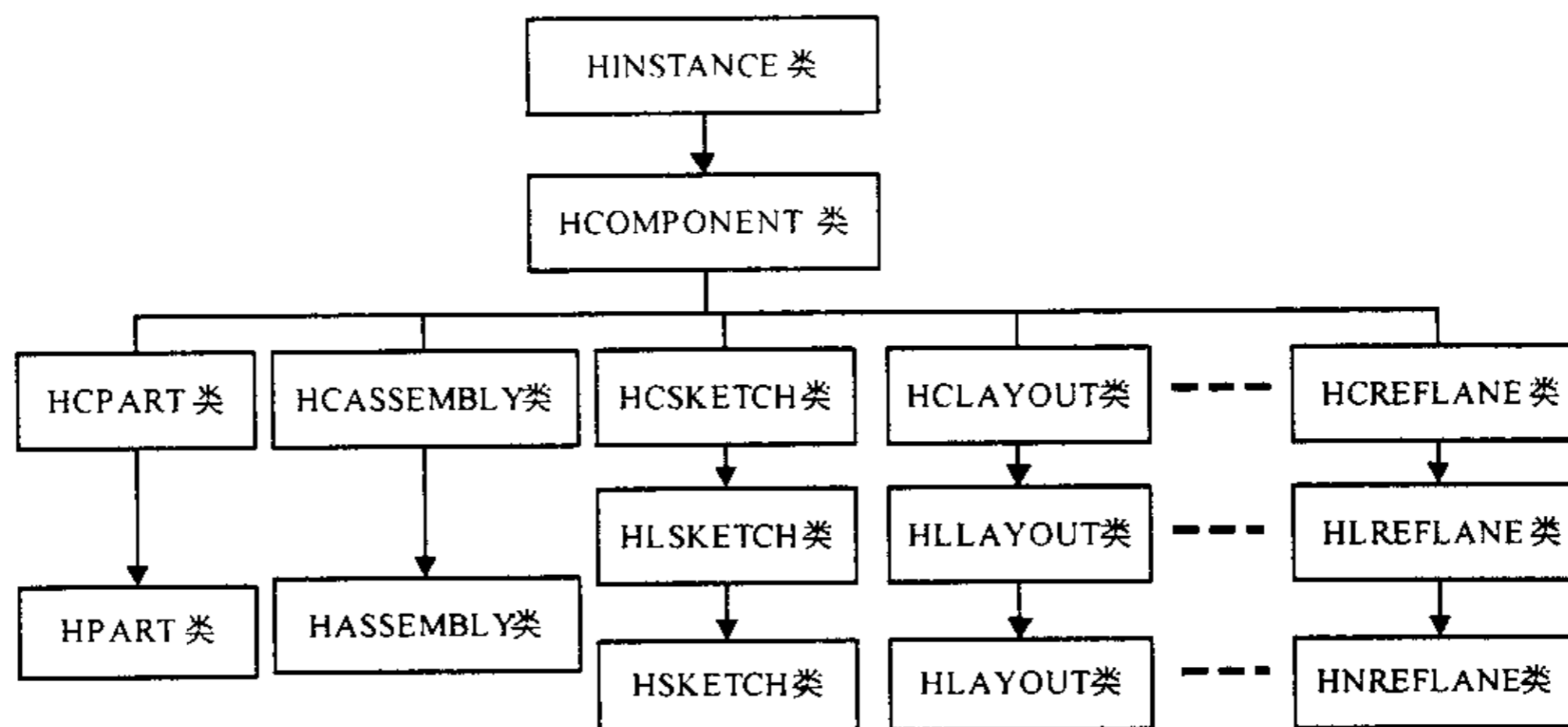


图 2.4 开放式装配模型总体结构

2.3.3 基于装配模型的布局设计框架

在布局设计过程中布局面、布局草图和布局零件的建立及求解是布局设计的核心。它们的共同作用可以实现布局设计，但是它们又相对独立，由于产品功能和形状的复杂性，以及布局设计各相关对象层次的复杂性，用一般的方法建立的布局设计模型异常复杂。不仅难以描述和维护设计意图及其布局表达方式，而且在布局草图向零件环境映射时也显得力不从心。这一切说明了布局设计中复杂性的内在原因是产品设计中布局方案的修改控制布局零件造型的重构。它也是传统的造型技术难以适应在产品周期中对设计意图经常性修改要求的主要原因。本节基于自顶向下的开放式装配模型，提出了开放式的布局设计框架，能较好的解决这个问题。

在开放式装配模型中，实元件结点是零件结点和装配结点，最基本元素是零件。而布局是产品概念设计在装配环境中的表达，通过装配环境的布局草图对零件详细设计进行控制，故布局结点是依附于装配模型实元件结点的。

如图 2.5 所示为基于装配模型的布局设计框架，方框表示结点 INS ，在图 2.5 中标有结点层次号。结点的层次关系实现布局设计过程，第一层结点 INS_0 是实现产品布局设计的布局结点；第二层 INS_1^* 是实现布局设计的布局草图结点，它们的关系 G 控制不同布局面的布局，每一个布局面的布局都是有此面上的布局草图元素来设计，布局草图中草图元素之间的关系 RE 则是实现每一个布局中各布局零件之间的配合关系。草图元素向零件环境进行映射形成零件基特征体。布局设计中布局面、布局草图等之间的联系是一种自顶向下的层次联系，布局设计能保证产品设计是一种功能概念设计的体现，实现布局设计对产品零件详细设计的控制，从而使产品零件的设计变更符合产品功能设计的变更。

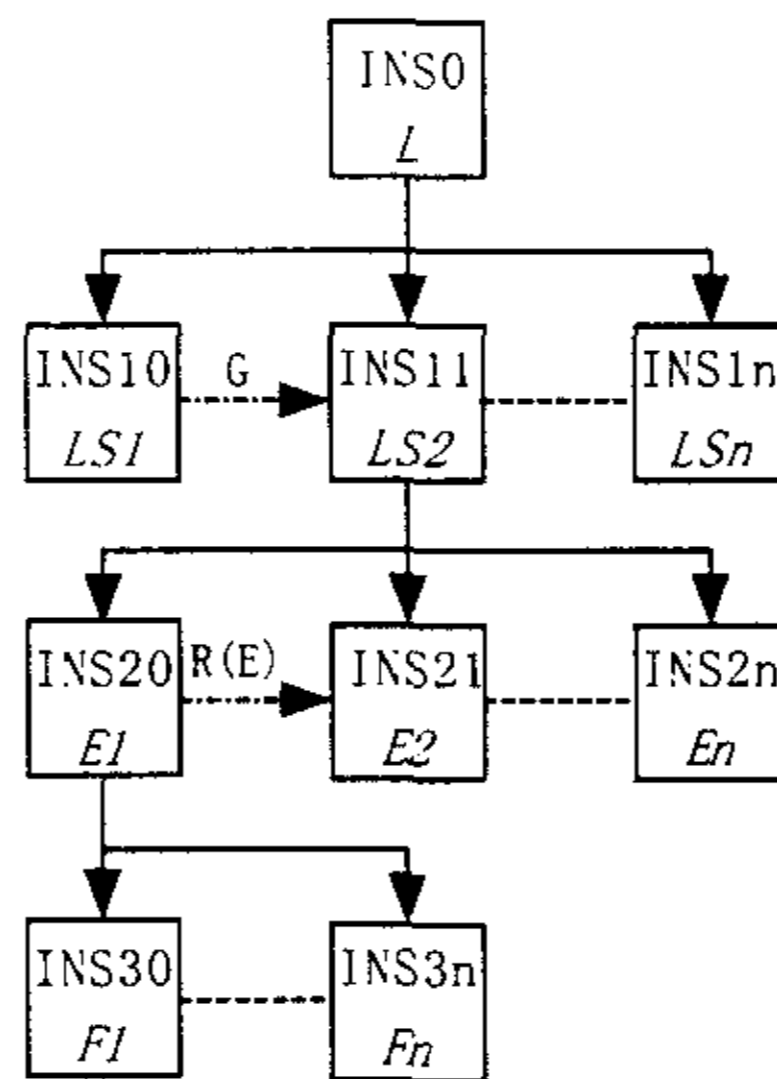


图 2.5 基于装配模型的布局设计框架

在这种模型中，只表达了布局设计的层次关系，而在装配体的设计中，装配体可以根据功能划分为子装配体、功能布局和零件，或者仅仅用功能布局。因此装配体的设计不全是单一的自顶向下的布局设计，而是布局设计和自底向上的零件设计的混合模式。这一模式可以用图 2.6 表示。

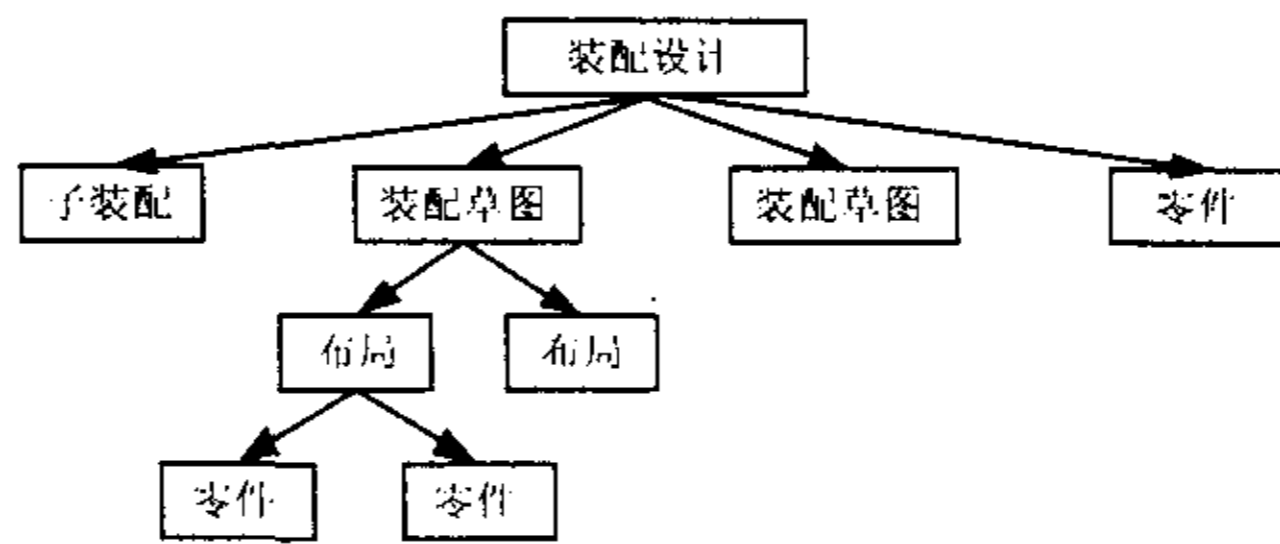


图 2.6 基于布局的装配设计图

2.4 布局求解闭环问题

由上一节布局的定义可知，布局方法是在装配环境中用二维草图的简单线条或者简单的草图元素控制三维几何约束中的复杂的零件体，三维零件体的详细设计须受到二维布局草图元素的控制；同时二维线条之间的共点、共线等约束关系控制着产品各零件之间的配合和位置关系，布局草图信息映射到零件环境对零件进行详细设计。因此当设计意图更改时，只需要修改二维布局草图则可以修改零件和零件之间的关系。在解决闭环约束问题上，用装配环境中的布局草图信息表达构成闭环约束的零件及其功能设计，因此用二维布局草图的闭环来表示三维空间闭环，实现闭环约束问题的简化，因此避免了数值迭代法对闭环约束求解的繁琐等缺陷。本节利用一四连杆机构来说明布局对闭环问题的求解。

如图 2.7(a)的简易四连杆机构，杆 L_1 为一固定杆，杆 L_4 与杆 L_1 夹角为关键设计变量 α ，初始值为 90° 。此四连杆机构的约束关系是相互关联的，工程上称为闭环机构，此问题的求解称为闭环约束问题求解。

在布局的求解方法中，首先由四杆机构的功能要求可知，四杆机构必须将四根杆零件装配成一整体，它是四个杆零件配合在一起的装配体；再次将这种功能分析的概念设计在三维设计系统中用布局草图体现出来，布局草图如图 2.7(b)所示，其中草图添加的约束为一种平面约束状态，因此每一个杆零件在空间须对六个自由度进行求解，到了平面则只需求解三个自由度，使三维闭环约束问题的求解得到简化。其中布局草图元素 $V = \{L_1, L_2, L_3, L_4\}$ 控制着四连杆零件的详细设计造型，而 $R(E) = \{(L_1, L_2), (L_2, L_3), (L_3, L_4), (L_4, L_1)\}$ 为共点约束等草图约束关系则控制布局零件之间的配合关系，固定点和点间的距离尺寸 $45.00mm$ 等控制布局产品各配合之间的位

置关系,因此布局草图的全部信息表达了四杆机构中各连杆零件的属性信息和配合关系信息。然后通过布局设计,将装配环境的布局草图映射到零件环境的基草图 $B = \{BS_1, BS_2, BS_3, BS_4\}$,用基草图 BS_i 线条控制第 i 个零件草图的构造,零件草图拉伸成杆零件的基特征 F_i ,在基特征 F_i 上完成杆零件的详细设计。装配环境和零件环境是两个独立环境,只有布局草图映射到零件环境中才可以对零件的详细设计进行控制。最后对杆零件进行详细设计,这样四杆机构的初始设计才得以完成。

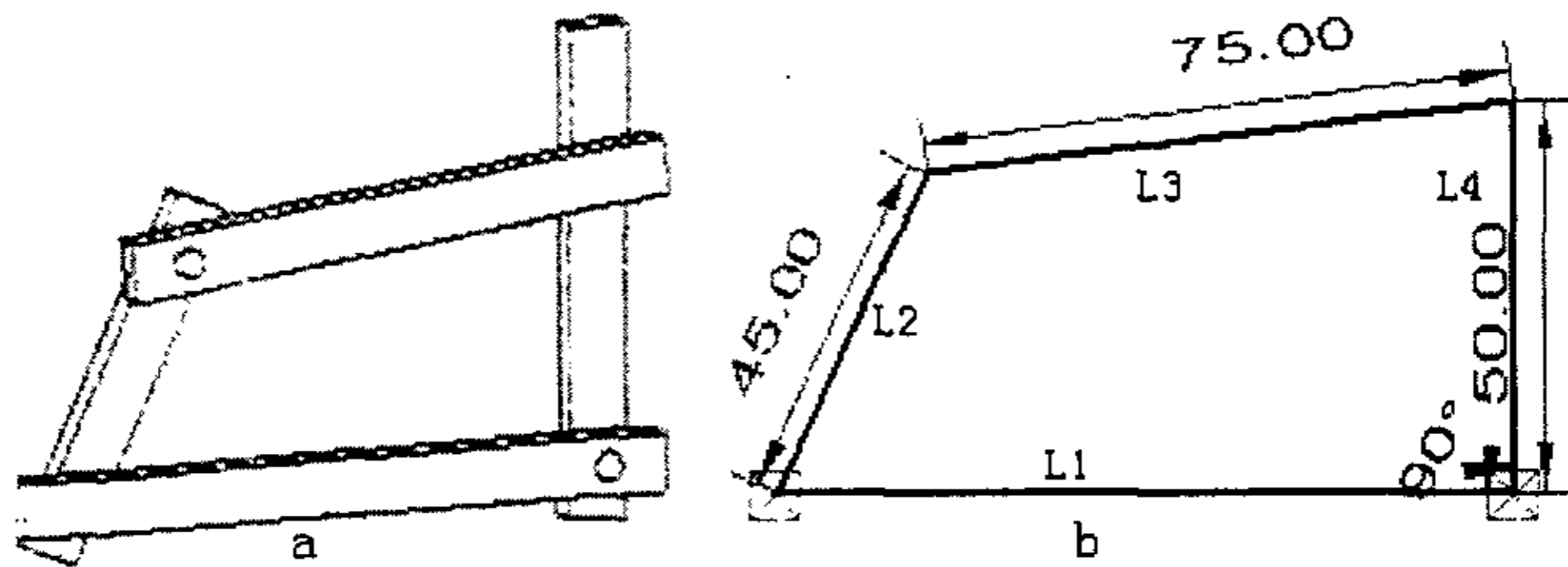


图 2.7 四杆机构及布局简图

在变量化设计系统中,变量技术的运用使四杆机构不同状态的获得成为可能,因此在变量化系统中设计四杆机构时,需要考察设计变量的变更导致整个设计产品的变更。布局的方法实现这种变更是采用修改布局草图中设计变量的尺寸 α 的值来实现。当 α 的值从初始 90° 变为 80° 时,四杆机构的布局草图如图 2.8(a)所示。

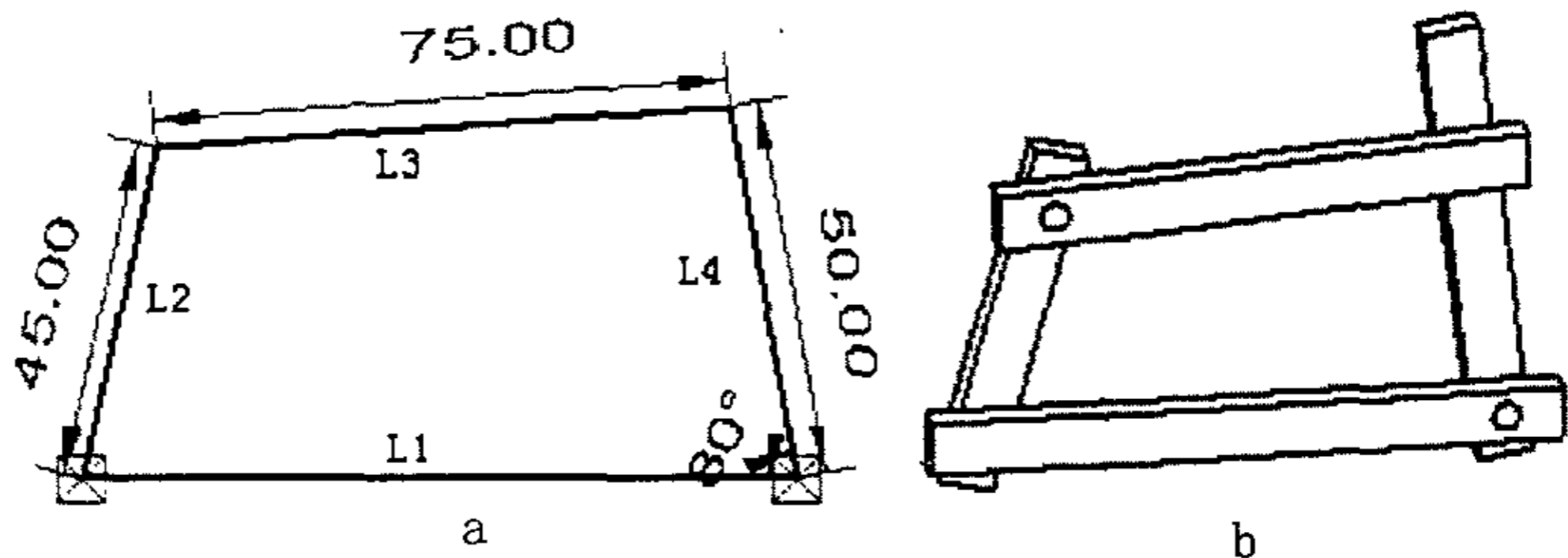


图 2.8 变化后的四杆机构布局及装配体

由图 2.8(a)知, 当设计变量 α 的值发生变化时, 布局草图中四线条之间的约束关系 $R(E)$ 没有发生变化, 配合位置和杆长度同样没有发生变化, 因此此变更的布局草图仍然可以控制四杆机构功能。完成对布局草图的修改, 也就完成了四杆机构在不同设计变量驱动下的设计。设计变量值 α 为 80° 时的四杆机构如图 2.8(b)所示。

2.5 本章小结

目前的三维变量设计系统主要采用几何推理和数值求解的方法来对三维几何约束进行求解, 但是求解闭环问题时, 都存在求解繁琐且不稳定的缺点。本章为了尝试解决这一问题, 提出了求解三维变量系统几何约束的布局的方法; 之后运用组件思想, 引入了利于布局设计的开放式装配模型, 提出了基于装配模型的布局框架, 成功的为解决三维变量设计系统的这一难题提供了一个解决方案; 最后给出布局求解工程中常见的闭环约束问题的实例。

3 布局关系模型

【本章摘要】为了实现布局设计，必须对布局对象进行建模。本章首先讨论布局相关的参考面、草图、布局等布局对象的定义，并给出各自的数据结构，在此基础上，给出各布局对象的相互关系定义。

3.1 引言

布局的功能是将产品的概念设计在三维系统中用草图表达出来，同时用以控制三维零件的设计，因此布局设计是在装配环境进行草图设计，零件环境进行产品零件设计，而传统三维变量设计系统中草图和零件设计均在零件环境完成。为了实现布局功能，因此需要实现装配层的布局面、布局草图以及它们与零件环境的布局零件之间的关系模型。

3.2 布局对象模型

3.2.1 布局面

在概念设计层的每一个简单草图都表达一定功能的零件配合体，而草图是二维平面图，任何一个草图都必须在一个参考平面上创造，同时参考平面的构造又可以引用它上层的草图元素。一般三维系统都会给定三个基准参考平面 XOY （俯视面）、 XOZ （主视面）、 YOZ （左视面），但是实际的产品设计中，往往存在一些不同于基准参考平面的非规则的参考平面，例如，平行于俯视面且两面的距离 $10mm$ 的参考平面，因此这些平面的定义就需要引用基准参考平面或者是其它参考平面上的草图元素，本文为了研究布局的需要，在 Intesolid2.0 系统中定义了以下非基准参考平面：

- 1) 偏移面：与一平面相距一定距离且平行于该面的平面。
- 2) 倾斜面：通过一条直边或参考轴并与一指定平面呈一夹角。
- 3) 三点定面：由不同在一个草图上的三点确定的平面。
- 4) 过点与平面平行：过一点并平行于一平面。
- 5) 过点与直线：过一点与一直线。
- 6) 过边端点与边垂直：过一点并与一直线或圆弧垂直。
- 7) 中面：与两平行平面间的距离相等。
- 8) 两点定面：两点定一平面，使该平面通过第一点，平面的法矢方向为从第一

点到第二点。

参考平面的数据结构如下：

```

HREFPLANE : public HASNODE
{
private:
    int          m_nId;           // Id 号
    HREFPLANE   *m_pPlane;      // 指向参考平面
    int          m_nType;        // 参考面的生成模式
    int          m_nComponent[3]; // 元件号
    int          m_nSketch[3];   // 草图号
    int          m_nElement[3];  // 元素号
    int          m_nRelation;     // 关系
    logical      m_nDirection;   // 方向
    positionm_Root;              //参考面 root(m_nMode =
                                //hrREFP_ROOTNORMAL 时)
    unit_vector m_uNormal;       // 参考面法矢(m_nMode =
                                //hrREFP_ROOTNORMAL 时)
    double       m_fOffset;      // 偏移值
    int          m_nSequence;    // 序列号
}
    
```

`m_pPlane` 是指向参考平面模块的指针，因为 `hhrefplane` 是装配环境的参考平面构造，它只是重载参考平面模块中的一些属性。`m_nType` 记录参考平面的生产模式，它们分别为偏移面等，并且在程序中将这面用宏对它们进行定义，

```

#define hr_as_OFFSET          0    // 偏移面
#define hr_as_ATANGLE        1    // 倾斜面
#define hr_as_3POINTS        2    // 三点定一平面
#define hr_as_P_PLANE_POINT  3    // 过一点并平行于一平面
#define hr_as_LINE_POINT     4    // 过一直线和一点
#define hr_as_V_CURVE_POINT  5    // 过一点且与一曲线垂直
#define hr_as_MIDPLANE       6    // 对称于两平行平面-中面
#define hr_as_2POINTS        7    // 两点定一平面
    
```

因此参考平面的类型用整型表示，`m_nSketch[3]` 记录构成参考平面的元素所在

的草图的 *id* 号, 因为对于三点定面而言, 三个点可能位于三个不同的草图, 因此草图号数组最大应该可以记录三个草图号, *m_nElement*[3] 表示元素号, 所选择构成参考平面的元素最多为三个, 因此数组 *m_nElement*[3] 必须开辟存储三个元素的空间; *m_nRelation* 是参考平面与其他元素的关系; *m_nDirection* 参考平面生成的方向, 与参考平面法矢相同的方向为默认方向; *Root* 参考平面在显示区的显示位置, 在系统中, 只有基准参考平面才会给定位置; *m_uNormal* 参考平面的法矢, 法矢用来判断参考平面的生成方向; *m_fOffset* 如果是偏移面则指定偏移面的偏移距离, 如果是倾斜面则指明倾斜面的倾斜角度。 *m_nSequence* 记录参考平面的序列号, 只有当参考平面个数超过三个基准面时, 此参数才有意义。

3.2.2 布局草图

布局草图, 装配层产品功能的表示, 支持功能设计的产品设计就是将功能进行分解为子功能, 再将子功能在装配环境用二维布局草图表达出来, 二维布局草图记载了有一定功能的产品的一些约束信息, 草图的几何形状来代表零件或者对布局零件构造的约束条件, 草图几何形状 (即草图元素) 之间的约束来表示产品中零件之间的配合、位置等的约束。

在 *Intesolid2.0* 系统中, 草图有单独的文档完成, 所以在装配层只需要将草图模块进行管理, 从而在装配环境实现草图操作, 因此在装配环境的草图构造中要有指向草图模块的指针, 而草图又是在装配环境中的参考平面上建立的, 因此需要记录草图所依存的参考平面的 *ID*, 为了维持布局草图和布局面等的历史关系, 每一个草图构造之后同样给一个序列号以记录草图的信息, 装配中草图的数据结构:

```
class HNSKETCH : public HASNODE
{
private:
    int      m_nId;           // Id 号
    int      m_nRefplaneId;  // 参考面的 ID
    HSKETCH *m_pSketch;     // 指向草图
    int      m_nSequence;    // 序列号
}
```

3.2.3 布局设计

自顶向下的设计就是将功能用布局草图勾画后, 再用包含有产品功能的布局草图来指导零件的详细设计, 因此布局设计是联接概念设计和详细设计之间必不可少的功能, 只有实现了布局设计才可以实现支持概念设计的产品设计。

布局设计的数据结构:

```
class DECL_ASM HLAYOUT : public HASNODE
{
private:
    int    m_nSketch;        // 草图编号
    int    m_nPart;         // 零件编号
    int    m_nPartSketch;   // 零件内对应草图编号
}
```

`m_nSketch` 记录装配环境中的草图编号, `m_nPart` 是装配环境中零件编号, 因为布局设计是实现装配草图对零件的设计过程的控制, 因此必须能记录零件内对应的草图的编号。本系统用 `m_nPartSketch` 表示。

3.2.4 布局零件

布局参考平面和布局草图的最终目的是实现底层的零件构造, 布局零件是满足布局草图的约束而生成的零件, 而自底向上的零件则是先有零件的形体属性, 再到装配环境中进行适应拼凑配合形成一装配体, 而布局零件则是先不清楚零件具体形状属性但是确定了零件之间的配合、位置等约束的基础上进行零件的形体属性的设计, 当布局草图构造以后, 发出布局设计的命令, 则可以用布局草图控制布局零件设计。而零件构造是在零件环境中进行, 装配环境则建立装配和零件之间的通信。

布局零件数据结构如下:

```
class DECL_ASM HCPART : public HCOMPONENT
{
private:
    HPART    *m_pPart;      // 零件指针
    HELEMENT *m_pElement;  // 指向零件编码
    int      m_nId;        // id 号
    int      m_nPartSource; //对零件来源的判断
}
```

以上数据结构中, `m_pPart` 是指向零件环境的指针, 用它来实现装配和零件环境的通信; `m_pElement` 实现零件环境中零件草图的编码与装配环境布局草图等的编码的一致性; `m_nId` 记录零件的历史 ID 号, 使零件环境的零件与装配环境中的布局处于同一历史环境中, 只有这样才可以使布局控制零件的详细设计; `m_nPartSource` 表明零件资源的来源, 它有三种来源: 一来自于零件环境的构造, 二来自于装配环境的

零件构造；三来自于布局设计。

3.3 布局关系建模

布局关系是实现布局设计的布局面、布局草图、布局设计和布局零件之间或者任意两者之间的关系；例如在布局面的构造中，三点定面必须引用一布局草图中的草图点，那么布局面和布局草图就存在一个历史先后的层次关系，三点定的面的属性就会随着拥有其草图点的草图的变更而改变。构成布局所需的对象之间的关系可以用图 3.1 进行定义。

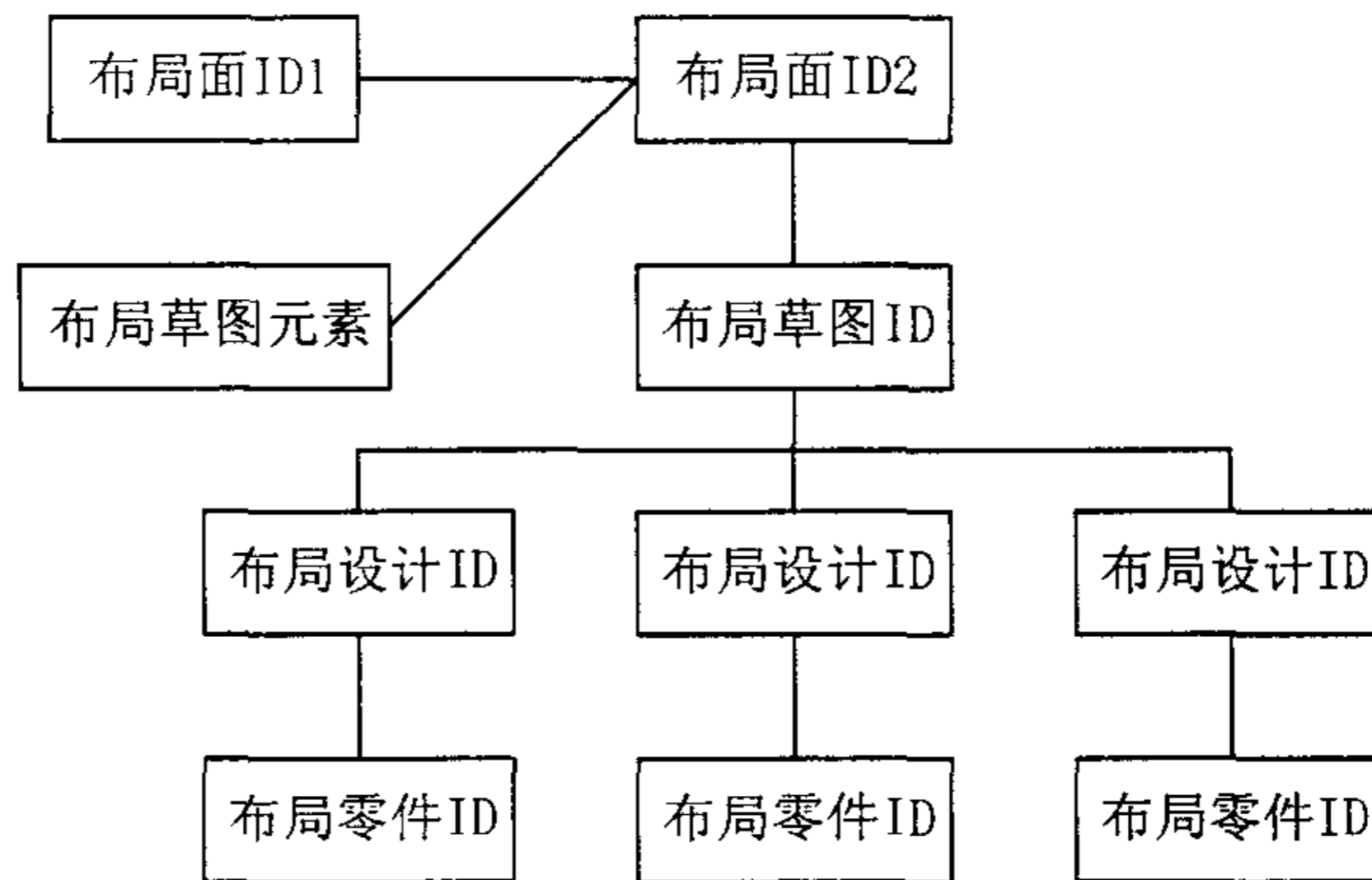


图 3.1 布局关系模型框架

在图中 ID_2 的布局面可能是由 ID_1 布局面和布局草图元素构造而成，但是三个基准参考面是由系统给定，因此它们的 ID 号也为系统给定，系统规定 XOY 的 ID 为 1， XOZ 的 ID 为 2， YOZ 的 ID 为 3。各布局对象 ID 号的编号原则是最先产生的布局对象的 ID 号最小，任何布局对象的变更或编辑定义都不能引用 ID 号大于本身 ID 号的对象或其中的元素。布局草图必须在一个布局面上进行构造，也就是产品功能方案须在一个平面上得以体现，而布局零件则是布局草图经过布局设计，根据不同的布局草图元素而产生不同的布局零件。

因为布局设计遵循 ID 号大的对象不可以修改 ID 号较小者，而零件是在布局关系

中的最下一层，也就是零件的 *ID* 必然大于产生此零件的布局草图的 *ID*，因此达到了布局草图对零件详细设计进行控制，而零件的详细设计只能反馈零件信息对产品布局设计方案进行评估并且提供修改的依据。

3.4 本章小结

布局设计是在装配环境进行草图设计，零件环境进行产品零件设计，而传统三维变量设计系统中草图和零件设计均在零件环境完成。为了实现布局功能，本章首先给出了布局所属对象布局面、布局草图、布局设计和布局零件的定义以及实现数据模型。然后详细论述了完成这一功能，布局对象应遵循布局草图控制布局零件详细设计、布局零件只可以反馈对布局草图检验信息的关系。只有这样才可以保证布局设计的自顶向下的特性。

4 布局求解过程

【本章摘要】本章首先依据布局设计的历史性，结合基于历史的编码/解码原理，提出了基于历史的布局编码/解码的映射机制，并给出布局设计历史维护算法实现了布局设计基于历史的重构，同时解决了自顶向下的零件造型的问题，最后本文给出了实例。

4.1 引言

三维变量设计系统不仅能够为零件设计提供设计工具，而且能够支持概念设计，体现设计意图，当某种原因导致概念设计的布局草图参数变化时，整个设计过程中的设计意图应该能够维护。布局草图的变化只是反映在装配体文档中，但是设计意图的维护则需要装配环境的布局草图控制零件环境中零件详细设计过程。这是三维变量设计系统所要解决的问题。本文采用基于布局构造历史的编码/解码原理和布局映射机制来实现对设计意图的体现和维护。

4.2 基于历史的布局及其性能

4.2.1 基于历史的布局

在包含布局设计的 CAD/CAM 系统中，需要嵌套多个建模环境，既存在着装配建模环境，又存在着零件建模环境。显然，每个环境的状态记录应该互相独立。这种独立性是建立在其中每一个实体的记录独立性的基础上的。对于三维变量设计系统中的每一个，理想的情形是可以遍历该对象的设计过程。对于该对象的每一次改变，都应该用一定的方式记录下来，这种记录不是全部记录，而是局部记录，即只记录对象的不同之处。从以上的分析可以看出，需要将多个该种类型的对象组合在一起共同组成某个对象的设计历史，如图 4.1 所示，同样，双向链表是这种方式的理想的选择，可以用一个前向指针和一个后向指针来实现。在此类对象中，要用指针记录“旧实体”和“新实体”，这里，“旧实体”和“新实体”都是对象相对于改变之前的对象内容的改变。可以用抽象实体对象来表示。因此设计历史(design_history)对象可表示为：

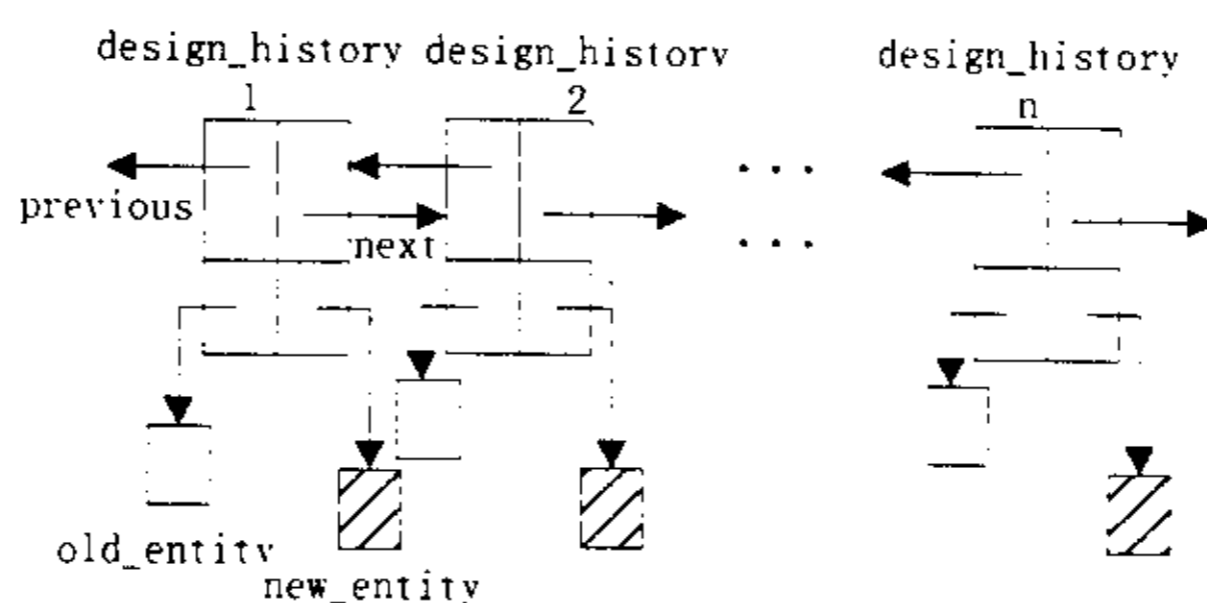


图 4.1 对象的设计历史示意图

```

ENTITY design_history;           //设计历史
    SUBTYPE OF (abstract_entity); //超类为抽象实体类
    design_history : previous;    //前向指针
    design_history : previous;    //后向指针
    old_entity     :abstract_entity //旧实体
    new_entity     :abstract_entity //新实体
END_ENTITY;
    
```

而布局设计则是在装配环境用布局草图体现设计意图，而对设计意图评估的零件的详细设计则是在零件环境完成，零件环境与装配环境为两个独立的设计环境，为了能够遍历产品的设计过程，必须记录布局设计的历史过程。

4.2.2 基于历史布局的基本性能

一个基于历史的布局需要满足如下两方面的性能要求。首先，要能够提供一个产品数据定义的工具。一般认为，产品定义的数据分为描述产品几何形状的几何数据及描述产品性能的非几何数据，包括与装配相关的数据和与零件相关的数据。其次，基于历史的布局应该能够支持设计全过程。对产品的设计过程进行维护，能对设计的下游应用如 CAM、CAPP 和 CAE 等等对设计的反馈作敏捷的反应，能维护产品在其生命周期中不断变更的产品信息。

因此，基于历史的布局应具备对设计过程的支持性。能表达产品在其生命周期中所蕴含的具体的不同产品信息，并能对产品信息变更作出敏捷的反应。

4.3 布局编码原理

编码就是将几何拓扑实体翻译成计算机可以识别并记录的文件来记录拓扑实体

的信息。

基于历史的三维变量化造型系统能够很好地定义尺寸和几何约束从而产生产品的设计变体(Design Variants)。为了产生产品的设计变体,设计师往往希望用自己提供的布局草图元素尺寸值和约束来定义产品模型并且希望能够有方便的编辑工具来重新定义产品模型。支持概念设计的的变量化建模系统的定义产品模型的过程是一个自顶向下的产品建模历史,每一步都包含了建模的操作、定义变量和属性。而变量化功能则提供能够在任何时候通过所存储的历史记录来自动重建产品的零件模型的能力,这种能力促成了产品设计变体的产生,而这正是变量化模型的主要目的之一^[51]。

基于布局的造型系统需要用布局草图的拓扑实体元素(面、边、点)上关联附件的信息来控制零件的设计,并且保证这些关联必须具有一致性,以保证在布局草图重建而导致零件模型重建时,拓扑关联信息能够保留下来并且不被丢失。典型的关联信息有:

- 1) 模型内部的特征依存。
- 2) 附加于面、边和点上的尺寸和注释。
- 3) 和模型关联的变量约束。

有了这些关联信息,模型就可以依据它们自动生成。但是模型通过约束的自动生成会导致许多语义上的问题^{[45][52]}。因此需要一个能对所引用的几何元素进行通用描述的机制,也就是入怀识别布局中的拓扑元素,即拓扑实体的编码(又称为拓扑实体的标识)。在此基础上可以保证在布局重建时这些拓扑元素依然能够被识别出来,并且能控制产品的重构。拓扑元素的编码和解码对于基于布局和约束的设计显得极为重要,通用的和健壮的编码和解码系统对于设计意图的维护起着决定性的作用,它是基于布局和约束的设计系统的灵魂。在此方面吴俊军博士^[53]作出了很大的贡献,并且将此技术在 intesolid2.0 系统中得以实现,为本文的布局研究帮助很大。

4.3.1 面编码

面的构造可能由其它面、草图中的元素等,如何识别构造面的这些几何拓扑实体以及如何在面重构时保持应该保持的拓扑关系的不变性,面的编码是识别面和取得面信息的必要条件。

本文采用的面编码是以原始特征体上的拓扑面作为编码的基础实体,而不是以零件体上的拓扑面作为编码的基础实体,更不是以原始特征体上或零件体上的拓扑点或拓扑边为编码的基础实体无疑是最佳的选择,这是因为:

- 1) 零件体上的拓扑边和拓扑点可能是无源拓扑边和拓扑点。
- 2) 零件体上的拓扑面都可以在原始特征体上找到对应的源。这样零件体的拓扑

面的编码就可以转化到原始特征体的拓扑面上。

- 3) 零件体和其上的拓扑实体处于不断的变化过程中,而原始特征体上的拓扑面却处于一种相对稳定的状态中。

以原始特征体上的拓扑面作为编码的基础实体,其优点在于:

- 1) 使得拓扑实体的编码有规律可循。那就是以原始特征面编码为基础,对零件上的拓扑面实体、点实体和边实体进行记录。
- 2) 编码转化到原始特征体上进行,无疑减小了问题的规模空间和复杂度。
- 3) 由布尔运算时面的拼合、分裂等带来的编码的传播处理转化到布尔运算的过程中解决,稳定地维护了编码的传播。
- 4) 为记录和解码的多解问题的解决奠定了坚实的基础。

4.3.2 草图编码

1) 草图编码

构成草图轮廓的草图元素分为点、直线、圆、圆弧和样条曲线五种类型,编码方案也相应地围绕这五种草图元素进行。草图元素的编码是其唯一的标识,这是一种一一对应的关系。

定义 4.1: 连接关系

如果两个草图元素之间有几何和拓扑上的连接关系,那么称这两个草图元素之间有连接关系。

定义 4.2: 虚连接关系

如果两个草图元素之间没有直接的几何和拓扑上的连接关系,但是它们之间的相对位置关系受到了整个草图轮廓的约束,那么称这两个草图元素之间有虚连接关系。

草图元素不是孤立存在的,它们必需通过连接或虚连接同其他的草图元素发生关系来共同构成草图轮廓。编码以草图元素的拓扑邻接关系为基础。为了严格保证草图的连接关系,对相连的草图元素,通过共用一个点草图元素来严格保证其相连接,而不用坐标点相等来连接,草图的编码就可以以点的编码为基础来进行,其余各种草图边都可以用构造它们的各种草图点的集合来表示。如图 4.2 所示的直线、圆、圆弧和样条曲线,虽然他们不能够决定所对应的草图元素的精确的几何表示,但却能够决定他们和其他的草图元素的拓扑邻接关系。

直线、圆、圆弧和样条曲线都是草图边,均由用不同数量的点构成,在拓扑邻接层次上,除了圆由一点标识外,直线,圆弧和样条都可用两点标识。

在草图中,点的编码必需保证每个点都有唯一的编码值,并且每个点编码值可以重用。编码值的确定采取由 1 开始,每增加一个草图点,其编码值增加 1 的方案,即:

$$Id_{point} = \{i | i = 1, 2, \dots, n\} \quad (4.1)$$

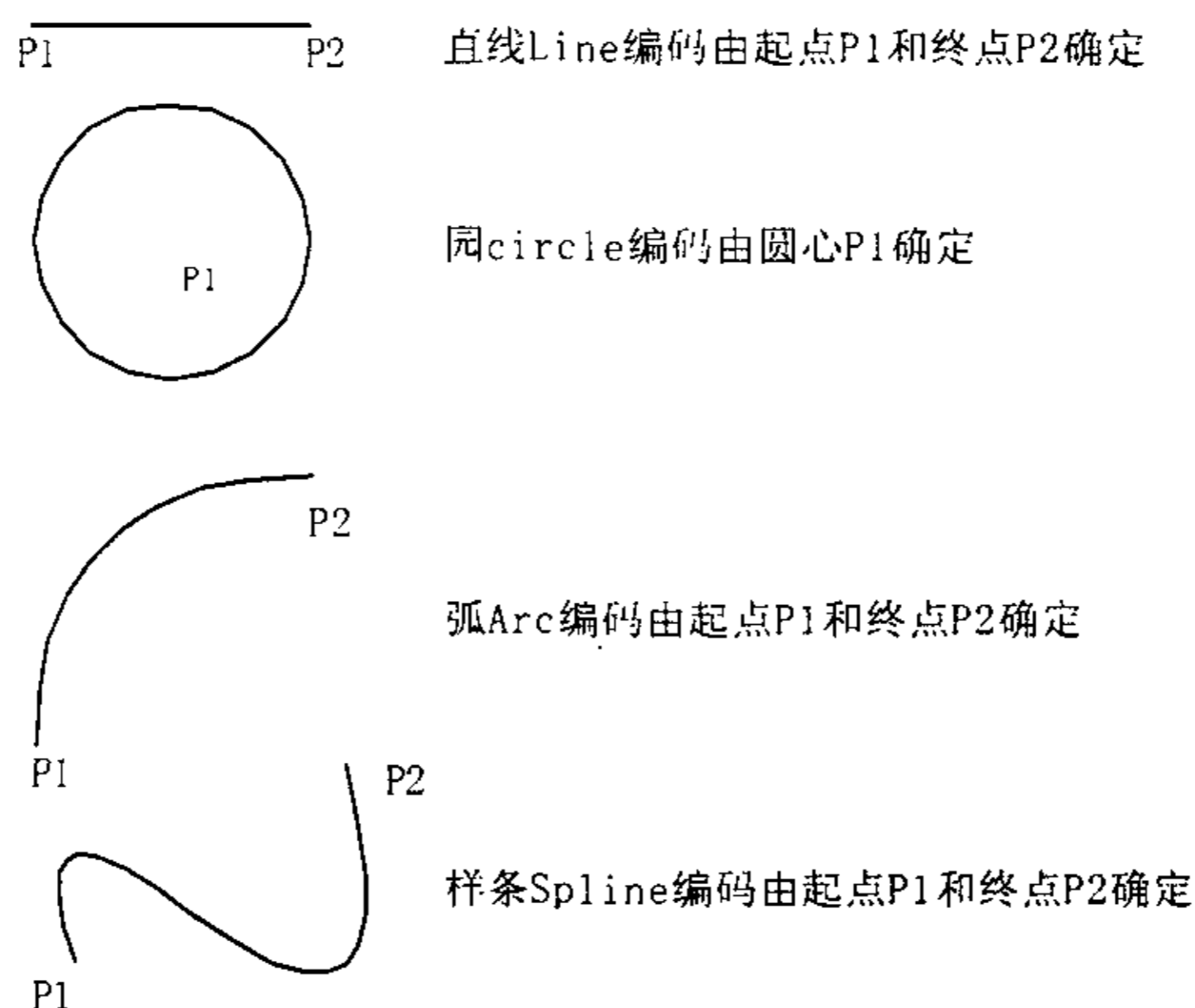


图 4.2 各种草图元素编码的确定

这样，基于以上事实，可以用最多两点组合实现草图轮廓的编码，即

$$ID_{element} = \{Id_{p1}, Id_{p2}\} \quad (4.2)$$

具体如下：

- 1) 直线：即线段，可由两 endpoint 唯一确定，因此直线的编码可以有两点的编码组合而成。最简单直观的方式是 $ID_{line} = \{Id_{p1}, Id_{p2}\}$
- 2) 圆弧和样条：的编码方法与直线一致。即 $ID_{line} = \{Id_{p1}, Id_{p2}\}$ ，
 $ID_{line} = \{Id_{p1}, Id_{p2}\}$
- 3) 圆的编码原理与上述基本一致，只是圆在草图轮廓中只有一个特征点，即圆心 $p1$ ，取 $Id_{p2} = 0$ ，这样圆的编码为： $ID_{circle} = \{Id_{p1}, 0\}$ 。

采取以上的编码方案后，图 4.3 中各个草图元素的编码如下：

$$Id_{p_i} = i, \quad i = 1, 2, 3, \dots, 9$$

$$ID_{line1} = \{Id_{p1}, Id_{p2}\} = \{1, 2\}, \quad ID_{line2} = \{Id_{p2}, Id_{p3}\} = \{2, 3\},$$

$$ID_{line3} = \{Id_{p3}, Id_{p4}\} = \{3, 4\}, \quad ID_{line4} = \{Id_{p5}, Id_{p6}\} = \{5, 6\},$$

$$ID_{arc1} = \{Id_{p4}, Id_{p5}\} = \{4, 5\}, \quad ID_{arc2} = \{Id_{p7}, Id_{p8}\} = \{7, 8\},$$

$$ID_{arc3} = \{Id_{p7}, Id_{p8}\} = \{7, 8\}, \quad ID_{spline} = \{Id_{p5}, Id_{p6}\} = \{5, 6\},$$

$$ID_{circle} = \{Id_{p9}, 0\} = \{9, 0\}.$$

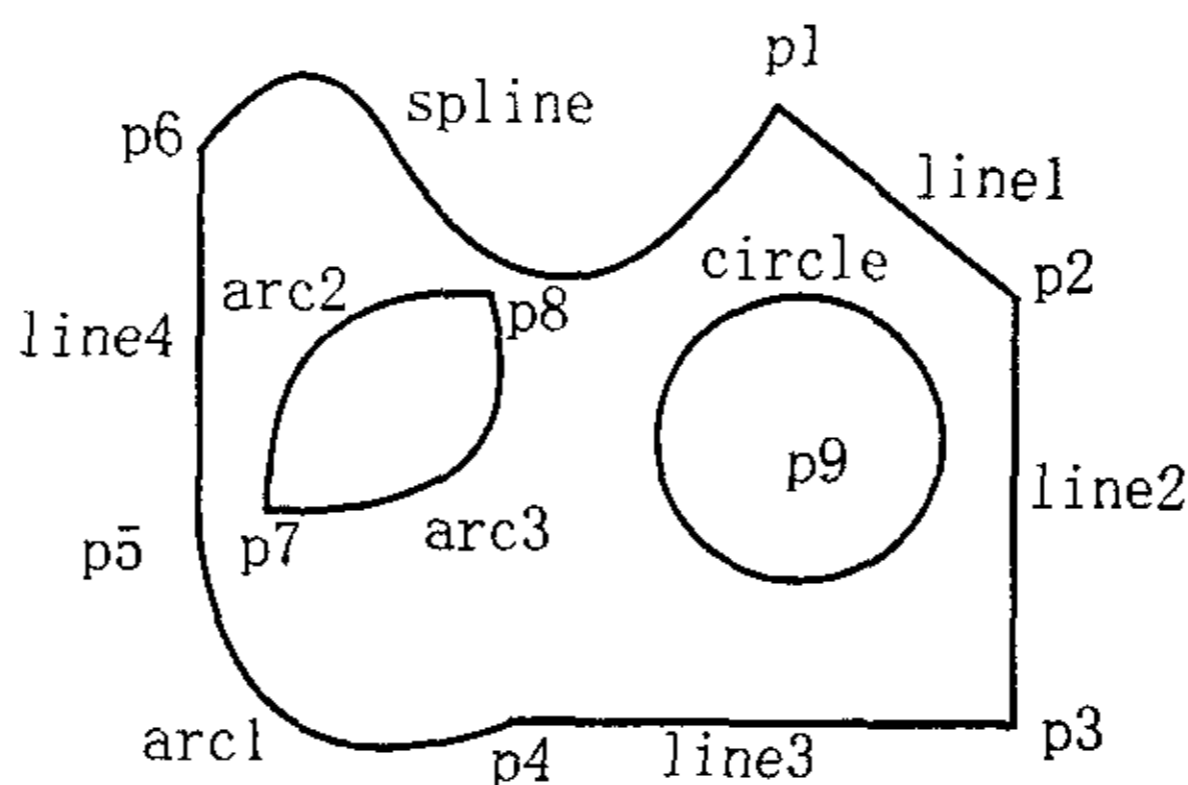


图 4.3 草图编码实例

2) 草图编码的修正

定义 4.3: 共享草图点问题

无法通过式 4.2 表示的编码来区分两条或两条以上共享草图点的曲线称为共享草图点问题。

草图轮廓中由直线、圆、圆弧和样条曲线中的某几种类型的曲线连接或虚连接而成。因而，它们可能共享一个或两个草图点，如图 4.3 中的圆弧 arc2 和 arc3，它们的编码都为 $ID = \{7, 8\}$ 。此时，无法通过编码来区分 arc2 和 arc3。共享一个草图点的情况表现为同心圆，而共享两个草图点的情况可以发生在直线、圆弧和样条三种曲线中的任意两种组合，或者是圆弧和样条两种曲线的同种组合，如直线和圆弧、直线和样条、样条和样条等等。如图 4.4 所示。共享草图点带来的直接的问题就是无法通过编码来区分草图元素。因此必需对草图元素的编码进行修正，采取对上述编码

$ID_{element} = \{Id_{p1}, Id_{p2}\}$ 加上另外一个索引项 Index, 即:

$$ID_{element} = \{Id_{p1}, Id_{p2}, Index\} \quad (4.3)$$

其中, Index 的取值从 1 开始, 对相同的草图点, 每发生共享草图点问题一次, 草图元素的 Index 值就增加 1。图 4.3 中的圆弧 arc2 和 arc3, 其编码修正后表示为:

$$ID_{arc2} = \{Id_{p7}, Id_{p8}, 1\} = \{7, 8, 1\} \text{ 和 } ID_{arc3} = \{Id_{p7}, Id_{p8}, 2\} = \{7, 8, 2\}。$$

同样图 4.4 中的直线 line、圆弧 arc、样条 spline 和弧 arc1 与 arc2 也可以有唯一的表示, 假设 $Id_{p1} = 1, Id_{p2} = 2$

以及 $Id_{p3} = 3$, 各草图元素编码表示如下:

$$ID_{line} = \{1, 2, 1\}, ID_{arc} = \{1, 2, 2\}, ID_{spline} = \{1, 2, 3\},$$

$$ID_{circle1} = \{3, 0, 1\}, ID_{circle2} = \{3, 0, 2\}$$

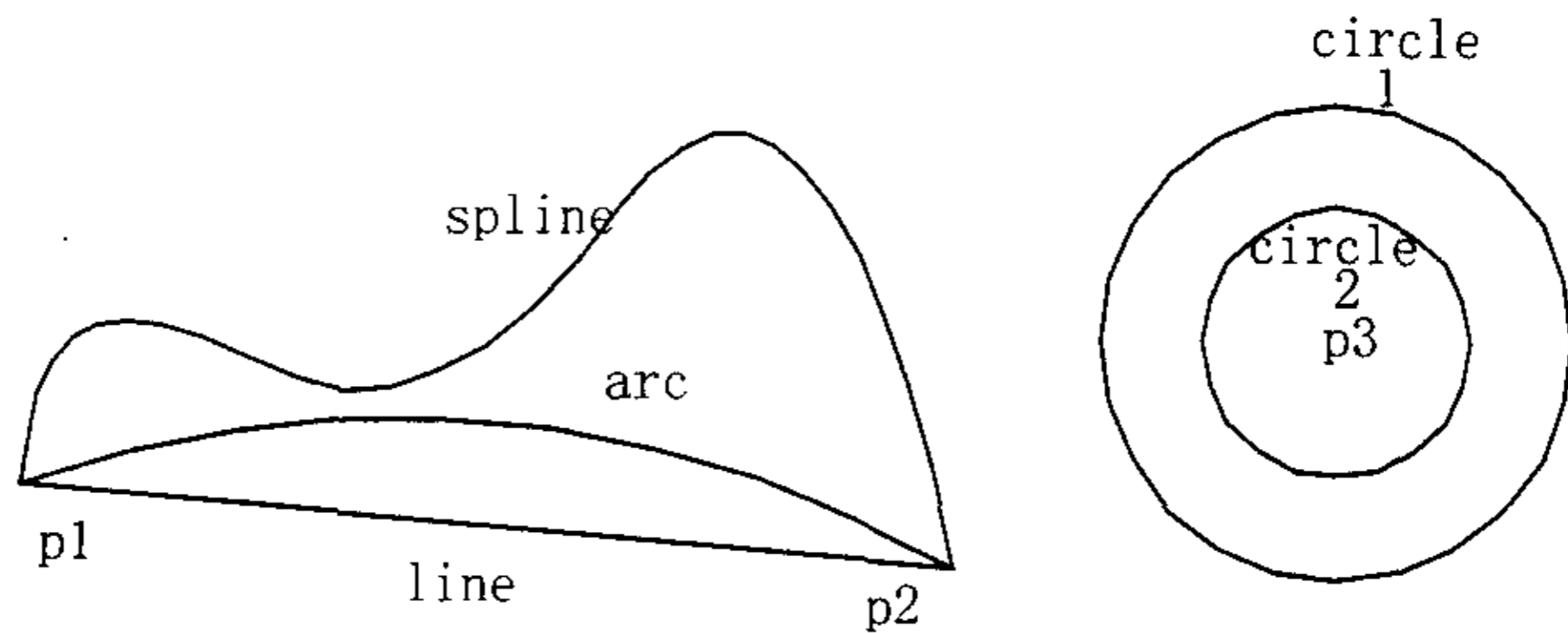


图 4.4 共享草图点问题

定义 4.4: 编码重生成问题

草图元素删除后, 又重新在原位置上生成类似的草图元素 (保证端点相同) 所带来的编码生成问题称为编码重生成问题。

编码重生成问题实质上是一个草图编辑中的编码一致性维护问题。简单来说就是草图边 edge1 被删除后, 又在相同 (近似) 位置生成另一条边 edge2, 怎样保证新生成的边 edge2 与原草图边 edge1 有相同的编码。边 $edge1(p1, p2)$ 变成了 $edge2(p1', p2')$, 由于草图边 edge2 是新生成的, 其编码是按前述方案重新生成的, 即

$ID_{edge2} = \{ID_{p1'}, ID_{p2'}, Index\}$, 通过紧凑编码可以使新生成的元素占据已删除元素的编码位置, 这样 $ID_{edge2} \cdot Index = ID_{edge1} \cdot Index$ 。此时, 新生成的边 $edge2$ 的编码可能出现两种情况, $ID_{p1'} = ID_{p1}, ID_{p2'} = ID_{p2}$ 或 $ID_{p1'} = ID_{p2}, ID_{p2'} = ID_{p1}$ 。这样, 在第一种情况下 $ID_{edge2} = ID_{edge1}$, 这正是理想的结果。而第二种情况出现了草图边编码的二义性, 这是要在编码中力图避免的问题。为了解决编码重生成问题, 必须对前面的编码方案再作一次修正, 即

$$ID_{element} = \begin{cases} \{ID_{p1}, ID_{p2}, Index\} & ID_{p1} < ID_{p2} \\ \{ID_{p2}, ID_{p1}, Index\} & ID_{p1} \geq ID_{p2} \end{cases} \quad (4.4)$$

3) 草图编码原则

在上述编码过程中, 有以下几条原则必须坚持。

- 1) 编码唯一性原则: 任何两个不同的草图元素, 其编码必然不相同。在进行草图编辑时也应该保证这一点。为了数据维护的简便易行, 在同一草图轮廓中, 系统不允许出现两个相同(重合)的元素, 当两元素重合时, 实际上就是同一元素, 这样就可以通过统一指针的方式保证数据的一致性。
- 2) 易于查询和大容量原则: 为了便于查询编码, 限制草图码编的长度为一个 32 位整形量, 对一个三元组可以用两个 12 位表示点的编码, 剩下 $32-12 \times 2=8$ 位作为剩下的索引值范围。这样点的编码范围为 $1 \sim 2^{12}-1$, 每个草图可容纳 4095 个草图点, 另外, Index 范围为 $1 \sim 2^8-1$, 共 255 个值, 可正确标识出 255 个同心圆或 255 个共用首末点的线条。由于草图设计针对的是三维系统中的二维截面图形, 一般比较简单, 所以这两个数字对绝大部分应用来说, 已经非常庞大了。
- 3) 编码紧凑原则: 编码必须紧凑, 也就是要保证有限的编码值得到充分的利用, 在上述式 4.4 三元组中, 对点的编码和索引值的选取都要紧凑。例如, 如果在原草图中共有编码为 1, 2, 3, 4 的点四个, 在编辑过程中删除了编码为 2 的点, 后来又加入了新点, 则新点的编码值应为 2 而不是 5, 同样的原理也适用于共用两端点的多条草图边的(直线, 圆弧或样条)索引的取值。这一条原则不仅能保证编码能有效地利用, 而且能够保证在草图编辑过程中最大限度地维护特征编码的一致性, 关于编码一致性问题, 将在下一节论述。
- 4) 编码的有效性原则: 通过编码值不仅能找到对应的元素, 而且能通过位运算

分离出其特征点的编码（对直线、圆弧和样条可得两 endpoint，对圆得其中心），这就为后续的操作提供了方便，这就是编码的有效性。

- 5) 编码的不变性原则：草图元素的编码是其唯一的标识，所以必需保证这个值在元素的生存期保持不变。这种原则在简化数据维护的同时也给编码提出了更高的要求。

在遵循以上原则的前提下，此编码方法所占字节少、容量大，并且保证了编码的唯一、完整和有效，这就为后续的操作（如参数化求解、特征操作）提供了方便。反过来，由于对编码有严格要求，这些后续操作也证实了编码的正确性。

4) 编辑过程中草图编码的一致性

草图编码的一致性在保证特征可编辑性以实现参数化特征设计的必要条件，草图编码是特征编码的基础，因此草图编码一致性的维护是十分必要的。在图 4.5 所示的零件中凸起特征 A 是由草图 SketchA 产生的，相应的特征面 FaceB 由草图边 LineB 产生，孔特征 C 依附于特征面 FaceB。如果设计者对特征 A 的形状不满意，重新编辑 SketchA，比如将草图 A 由四边形修改为五边型，若系统不能保证草图边编码的一致性，则修改后 $ID_{newlineB} \neq ID_{oldlineB}$ ，那么 $ID_{newfaceB} \neq ID_{oldfaceB}$ ，重构此零件时，特征 C 重新生成时要通过查找 $ID_{oldfaceB}$ 确定其基准平面，由于此时 FaceB 的编码已改变，特征管理器将永远无法找到特征 C 的基准面，这样特征 C 就永远丢失了，那么设计者的意图没有得到应有的维护。

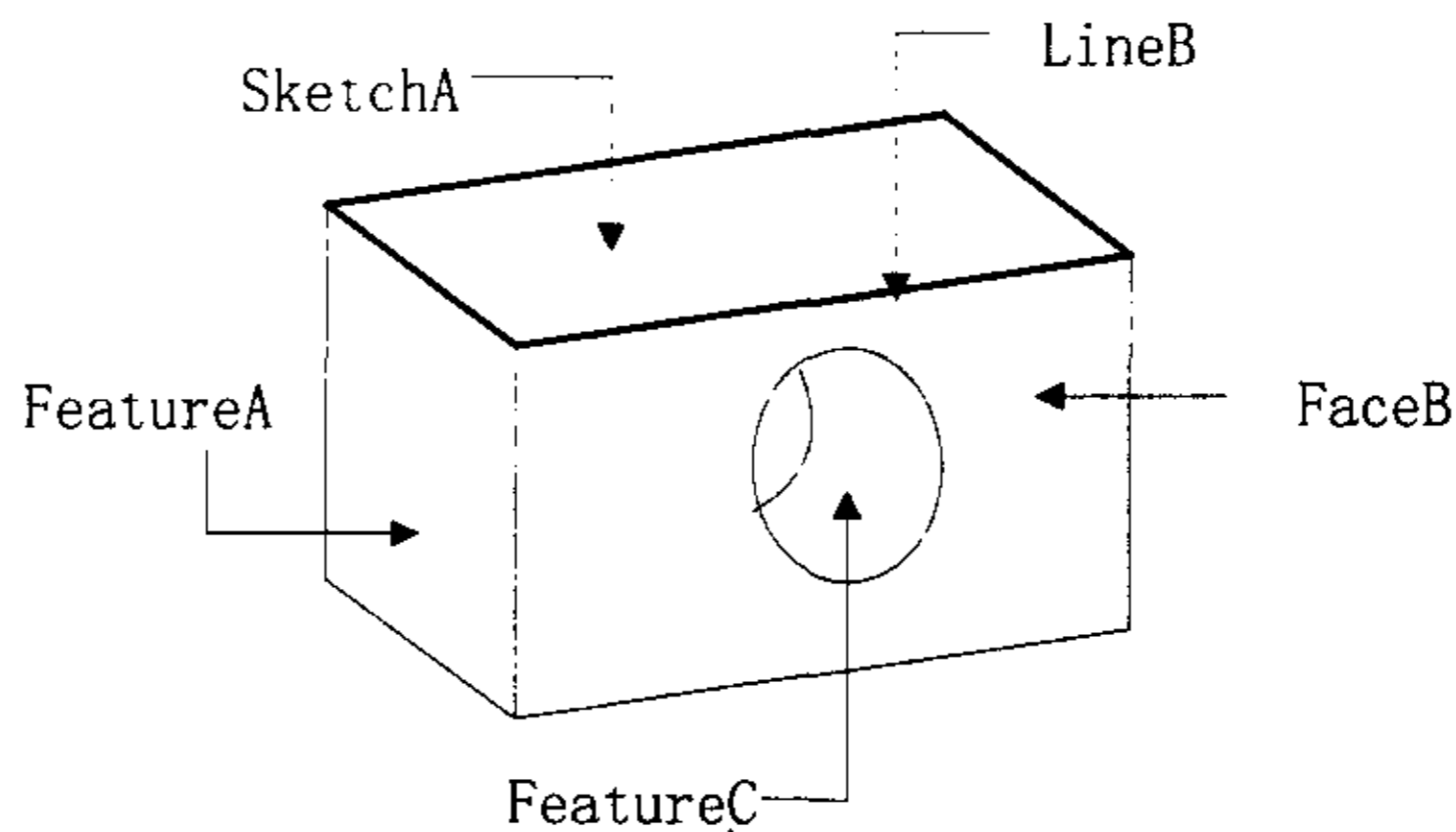


图 4.5 编码的一致性

本节将分析在不同的编辑情况下草图边一致性的维护。从编辑对草图的拓扑所产生的后果来看，草图编辑可分为不变拓扑的编辑和变拓扑的编辑。

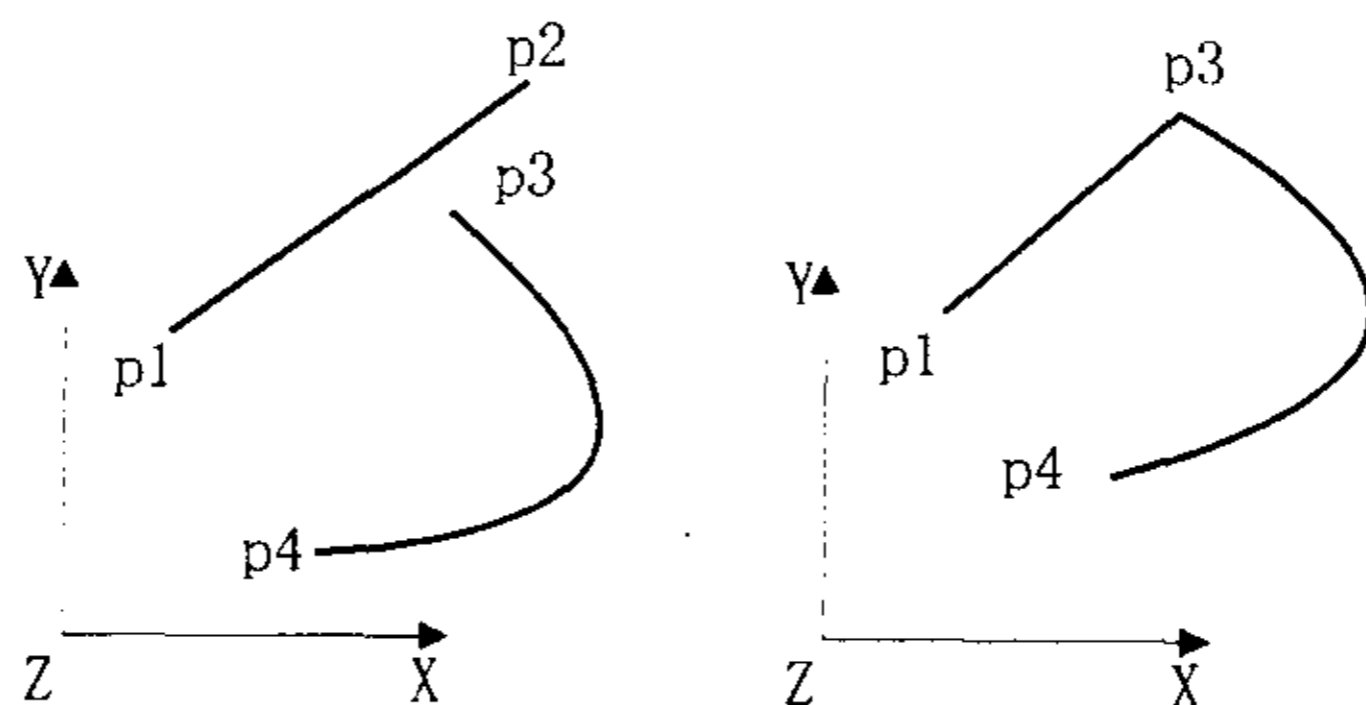


图 4.6 草图编码一致性

- 1) 不变拓扑的编辑指通常情况的参数化, 如添加约束修改尺寸, 拖动部分草图等, 这种情况下, 草图的任何元素的编码都不会改变, 这是因为草图元素只是修改了几何数据, 而其拓扑数据并没有修改, 而几何数据的改变是不会影响草图元素的编码值的。
- 2) 变拓扑的编辑, 这种情况下某些草图边的端点发生了变化, 如图 4.9 所示, 直线 line 的端点为 p1 到 p2, 用户拖动直线 line, 使得 P2 与草图上另一有效点 P3 重合, 这样直线 line 的端点变成了 p1 到 P3, 而点 P2 则自动删除。为了维护边编码的一致性, 这种情况下对直线 line 不进行重新编码, 只修改其组成端点, 这也是编码不变性原则的体现。

4.3.3 基于草图编码的零件特征编码

如图 4.7 所示的是一典型的广义轮廓扫, 形成的形状特征的面有起始面(StartFace), 扫成面(SweepFace)和终止面(EndFace)三种类型。设二维轮廓草图为 P, element 为其中的某草图轮廓边, 路径为 Path, trajectory 为其中的某个路径片断, 由 element 和 trajectory 决定的形状特征面 $F(\text{element}, \text{trajectory})$ 在此形状特征中可以用 FeatID_P , $\text{ID}_{\text{element}}$, $\text{FeatID}_{\text{Path}}$ 和 $\text{ID}_{\text{trajectory}}$ 来确认。其中 FeatID_P 和 $\text{FeatID}_{\text{Path}}$ 分别是二维轮廓草图和路径的编码, 由于二维轮廓草图类和路径类均由特征类派生, 因此它们的编码 FeatID_P 和 $\text{FeatID}_{\text{Path}}$ 等于其特征标识。而 $\text{ID}_{\text{trajectory}}$ 确定的方法和二维轮廓边的编码的一样。故基于草图的特征的扫成面的编码为以下的五元组:

$$C_{\text{sweepface}}(\text{FACE}) = \{ \text{FeatID}_{\text{sweep}}, \text{FeatID}_P, \text{ID}_{\text{element}}, \text{FeatID}_{\text{Path}}, \text{ID}_{\text{trajectory}} \} \quad (4.5)$$

其中 $\text{FeatID}_{\text{sweep}}$ 为此广义轮廓扫的特征标识。

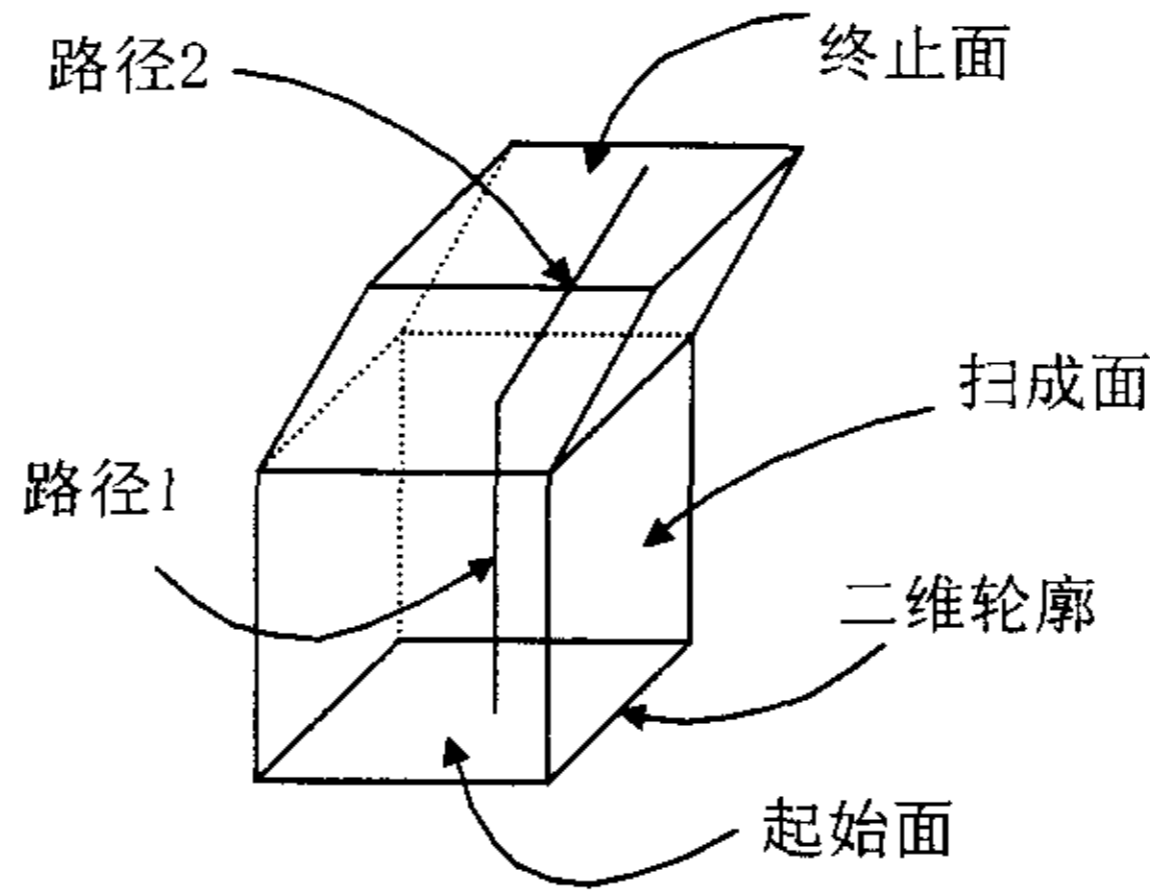


图 4.7 典型的广义轮廓扫

起始面和终止面是两个可以快速确定的面，因为对于绝大部分广义轮廓扫而言，都有起始面和终止面，因此它们的编码具有广泛的意义。对于起始面，取 $ID_{element}$ 的值为 -1，对于终止面，取 $ID_{element}$ 的值为 -2，而它们的 $FeatID_p$ 、 $FeatID_{path}$ 和 $ID_{trajectory}$ 都为 0。结合起始面和终止面，基于草图的特征面的编码为：

$$C_{Sweep}(FACE) = \begin{cases} \{FeatID_{Sweep}, 0, -1, 0, 0\} & FACE = StartFace \\ C_{SweepFace} & FACE = SweepFace \\ \{FeatID_{Sweep}, 0, -2, 0, 0\} & FACE = EndFace \end{cases} \quad (4.6)$$

对于线性轮廓扫，其是沿轮廓的法矢方向扫描，因此 $FeatID_{path}$ 和 $ID_{trajectory}$ 的值为 0，故线性轮廓扫的特征面的编码为：

$$C_{Extrude}(FACE) = \begin{cases} \{FeatID_{Extrude}, 0, -1, 0, 0\} & FACE = StartFace \\ \{FeatID_{Extrude}, FeatID_p, ID_{element}, 0, 0\} & FACE = SweepFace \\ \{FeatID_{Extrude}, 0, -2, 0, 0\} & FACE = EndFace \end{cases} \quad (4.7)$$

对于旋转轮廓扫，其是绕轮廓所在平面的某根中心轴 $axis$ 旋转，因此 $FeatID_{path}$ 和 $ID_{trajectory}$ 的值分别为 $FeatID_p$ 和 ID_{axis} ，值得注意的是，当旋转 360 度时旋转轮廓扫将没有起始面和终止面，故旋转轮廓扫的特征面的编码为：

$$C_{Revolve}(FACE) = \begin{cases} \{FeatID_{Revolve}, 0, -1, 0, 0\} & FACE = StartFace \\ \{FeatID_{Revolve}, FeatID_p, ID_{element}, FeatID_p, ID_{axis}\} & FACE = SweepFace \\ \{FeatID_{Revolve}, 0, -2, 0, 0\} & FACE = EndFace \end{cases} \quad (4.8)$$

4.3.4 历史过程编码

布局求解是一种基于历史的求解，因为布局设计各对象之间的历史性决定了布局

草图对零件详细设计的控制和指导作用,也决定了零件设计对布局草图对产品功能表达的无误性。为了实现布局各对象之间的历史控制关系,布局的历史过程也需要进行编码。

布局历史过程编码的原则:

- 1) 依照布局对象产生的先后顺序进行编码;
- 2) 布局对象的编辑定义不改变它的历史编码;
- 3) 布局对象的编辑定义不可以引用它后面的历史对象;
- 4) 某一布局对象的删除,依附于此对象的编码也随之删除。

按照以上原则,用 ID 号来为布局对象产生的历史过程进行编码。

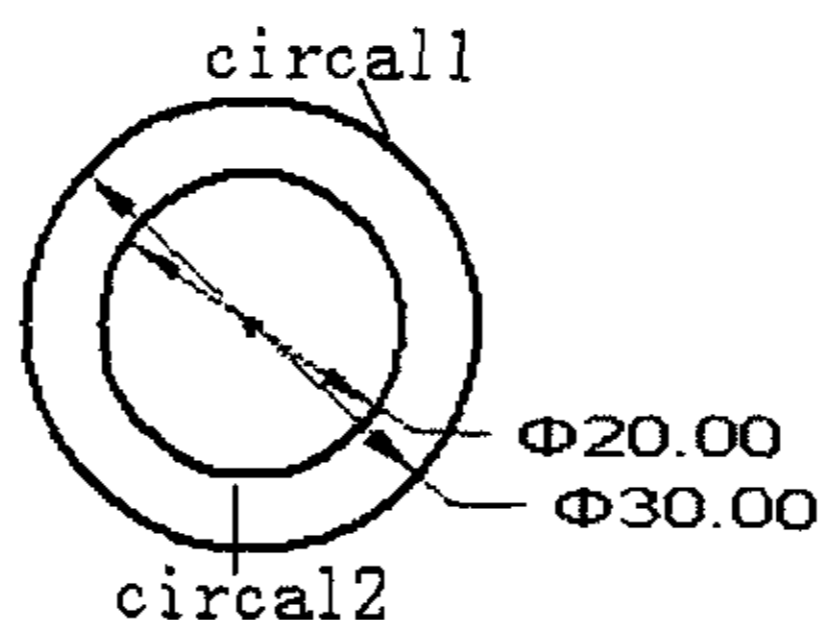
4.4 布局编码的映射机制

布局面和草图的编码记录最终是为了实现布局控制零件环境的详细设计,但是布局是在装配环境中进行编码,而零件却又是在零件环境中进行编码,它们的记录是独立的,如何实现装配环境中的布局来控制相对零件环境中的零件设计?本文提出布局编码向零件映射机制,建立布局草图与零件之间历史性。映射可以定义为一个平面或空间的物体按照一定的变换到另一个平面或者空间,但是此物体的内部组织和内部组织之间的关系却没有发生任何变化。

基于布局的映射需要以下功能:

- 1) 布局面编码到零件面编码的映射;
- 2) 布局草图所有编码到零件草图编码的映射;
- 3) 零件的详细设计结构到装配环境的映射。

例如简单的轴和轴孔配合的产品布局草图在装配环境中的构造如图 4.8 所示。



4.8 布局草图

布局草图依托的参考平面在历史树上的记录为: $ID_{ref} = 3$, 那么布局草图的历史

纪录为 $ID_{skt} = 5$ 布局草图元素（两个同心圆）编码分别为： $ID_{circle1} = \{1,0,1\}$ ；

$ID_{circle2} = \{1,0,2\}$ ，因此记录的草图两元素的基于历史的编码为：

$$INS_{shaft}(Layout) = \begin{cases} ID_{ref} = 3 \\ ID_{skt} = 5 : \begin{cases} ID_{circle1} = \{1,0,1\} \\ ID_{circle2} = \{1,0,2\} \end{cases} \end{cases} \quad (4.9)$$

装配环境的布局草图要映射到零件环境为零件设计所引用，根据布局对映射的要求，必须使布局面和布局草图都能够不改变装配环境的布局设计历史关系地映射到零件环境中，因为在零件环境中的设计历史与装配环境的设计历史是相互独立的，装配布局到零件环境中，设计历史过程必须从头开始记录，但是布局面和布局草图的历史相对顺序到零件环境应该保持一致，所以采取的映射机制使布局面总是映射为零件环境中的 XOY 平面，其上的草图映射为第一个草图，其在零件环境的历史位置为三个基准参考平面之后即 $ID_{skt} = 4$ 由上可知，轴与轴孔的布局映射到零件环境它的历史关系和元素编码如下：

$$SKT_{shaft}(Part) = \begin{cases} ID_{ref} = 1 \\ ID_{skt} = 4 : \begin{cases} ID_{circle1} = \{1,0,1\} \\ ID_{circle2} = \{1,0,2\} \end{cases} \end{cases} \quad (4.10)$$

得知轴和轴孔配合在零件环境的草图形状，在产品意义上，轴和轴孔的配合必须是由一个轴和带有中孔的轴套两个零件，一个布局草图可以控制多个布局零件的设计，因此我们建立基于草图的零件特征编码完成零件的生成，对于轴套是基于 circle2 而生成的零件，轴则是基于 circle1 和 circle2 相交的草图环生成，因此它们共用了 circle2 元素，必然 circle2 的编码即控制轴也控制轴套，以轴套的孔的生成与轴的生成的控制元素相同来实现轴和轴套的配合。由零件环境中的基于草图的特征编码的通用表达式：

$$C_{sweepface}(FACE) = \{FeatID_{sweep}, FeatID_p, ID_{element}, FeatID_{path}, ID_{trajectory}\} \quad (4.11)$$

其中 $FeatID_p$ 为二维轮廓草图的历史编码， $ID_{element}$ 为其中的某草图轮廓边编码， $FeatID_{path}$ 二维路径编码， $ID_{trajectory}$ 为其中的某个路径片断编码， $FeatID_{sweep}$ 为此广义轮廓扫的特征标识。

起始面和终止面是两个可以快速确定的面，因为对于绝大部分广义轮廓扫而言，都有起始面和终止面，因此它们的编码具有广泛的意义。对于起始面，取 $ID_{element}$ 的

值为-1, 对于终止面, 取 $ID_{element}$ 的值为-2, 而它们的 $FeatID_p$, $FeatID_{Path}$ 和 $ID_{trajectory}$ 都为 0。而又因为本例为线性扫轮廓, $FeatID_{Path}$ 和 $ID_{trajectory}$ 的值为 0, 结合起始面和终止面, 可知轴的设计编码为:

$$SHA_{Extrude} = \begin{cases} \{FeatID_{Extrude}, 0, -1, 0, 0\} & FACE = StartFace \\ \{FeatID_{Extrude}, ID_{skt}, ID_{circle2}, 0, 0\} & FACE = SweepFace \\ \{FeatID_{Extrude}, 0, -2, 0, 0\} & FACE = EndFace \end{cases} \quad (4.12)$$

在工程加工中轴套一般都是有两个特征构成, 因此轴套编码为:

$$SLE_{Extrude} = \begin{cases} CIR1_{extrude}(FACE) = \begin{cases} \{FeatID_{Extrude}, 0, -1, 0, 0\} \\ \{FeatID_{Extrude}, ID_{skt}, ID_{circle1}, 0, 0\} \\ \{FeatID_{Extrude}, 0, -2, 0, 0\} \end{cases} \\ CIR2_{extrude}(FACE) = \begin{cases} \{FeatID_{Extrude}, 0, -1, 0, 0\} \\ \{FeatID_{Extrude}, ID_{skt}, ID_{circle2}, 0, 0\} \\ \{FeatID_{Extrude}, 0, -2, 0, 0\} \end{cases} \end{cases} \quad (4.13)$$

其中 $ID_{skt} = 4$, 由此历史编码可得到由布局控制的轴和轴套的零件的详细设计信息, 在装配环境中我们可以看到由布局控制而生成的零件装配体。

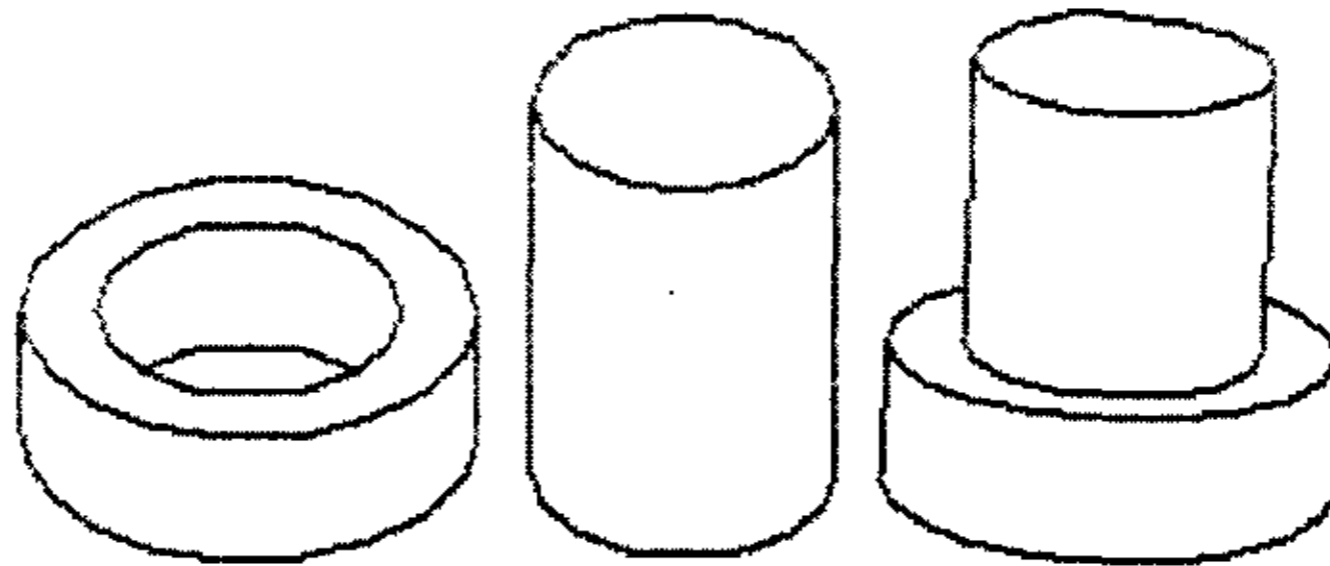


图 4.9 利用映射机制产生的布局装配体

4.5 布局求解

由上一节的布局历史编码映射机制我们知道, 在布局的设计历史过程中, 布局面、布局草图、布局设计和布局零件之间存在层次关系, 布局草图需要在布局面上生成, 而只有完成了布局草图的设计才可以进行布局设计, 而布局设计则连接布局草图和布局零件。这些层次关系都是自顶向下的控制产品的设计过程, 工程设计中的功能需求

决定了产品的设计, 产品设计中功能的表达都是通过零件之间的配合关系得以实现, 在上例中表现为轴和轴套这两个零件的装配体, 用布局草图实现轴和轴套的概念设计, 真正的自顶向下的设计都是支持概念设计的一种设计过程, 这就要求布局草图能对零件的详细设计进行控制, 因此通过布局设计实现布局草图到零件草图的映射, 用映射到零件环境的布局草图实现和控制零件的设计, 并且在装配环境中保存零件设计的信息。在此过程中, 因为零件草图是由布局草图映射而成, 因此在零件环境禁止对布局草图的修改, 但是布局草图的修改则会影响零件草图的修改, 从而导致零件的更改, 正是布局的这种层次关系维护了自顶向下的产品设计的历史过程。

4.5.1 设计意图的表达

本文布局的设计是通过二维草图的设计实现。定义一设计意图 P , 可有一系列的组件及其之间约束关系表达 $P = Intent(\sum Com_i, \sum Con_i, \sum Dim_k)$, 而设计者最初的表达方法多时进行平面上的粗略轮廓表达。三维设计系统由于具有二维草图设计工具, 能够维护初步设计轮廓 Com 及其之间的约束关系 Con, Dim , 因而能够有效地表达设计意图 P 。

4.5.2 布局的建立

自顶向下的产品设计是由概念设计来控制产品的详细设计, 而概念设计在三维系统中以布局的形式表达, 因此实现自顶向下的产品设计的关键就是在三维设计系统中建立产品概念设计的布局。

在本系统的层次结构中, 布局面和布局草图都是与布局设计在同一层上, 因此建立布局必须将布局面、布局草图与布局设计结合起来为指导产品详细设计的需要。布局草图中的几何约束为布局设计的零件之间提供装配体的配合约束, 而变量约束则控制布局零件的属性尺寸和各零件之间的位置关系。

4.5.3 布局历史过程的维护

产品设计过程都是表达设计意图 P 的过程, 在变量化设计系统中, 通过设计---再设计的反复变量求解来实现 P , 布局设计中变量的改变要以不破坏设计意图实现的历史过程为准则。因此需要对布局历史过程进行维护。

当某一布局历史结点发生变化时, 那么那些引用了这一历史结点的后续结点的属性则需要用此改变的布局结点的属性来进行重新构造, 这样有可能会破坏此后续结点的编码和它在零件环境中的映射关系。因此维护布局历史过程不仅需要维护布局的历史构造过程, 也需要维护各历史对象的编码以及它们的映射机制。

上一节提到的编码原理不仅能够为拓扑实体提供唯一的、一致的特征编码，而且通过对编码进行解码，也可以反映出设计过程中拓扑实体间的依存关系以及产品最终或中间几何体上的几何元素和初始特征体（Proto Feature）间的映射关系。这些关系不仅为模型重构所必需，而且也是建立和维护产品信息关联所必不可少的。

布局设计采用这种编码/解码原理，每一个布局都给一个历史ID，每一个布局对应的草图与布局一起编码，这样就获得了草图和布局之间唯一的、一致的编码。并且采用布局编码的映射机制，将布局编码映射到零件环境进行零件的详细设计。这种由布局编码控制的布局零件同样在装配体环境进行编码。而布局的解码原理则是将编码的布局设计重新翻译成装配环境的几何拓扑实体。当布局设计对象中的变量修改时。依据编码的ID按照深度优先原则进行搜索，查找到引用此设计变量而构造的布局设计对象ID，并将他们按照设计变量的更改进行重构，从而得到维护设计意图的布局设计变体。

维护设计历史，实现设计意图，构造以下搜索重构算法：

```
PROCEDURE Rebuild(Variable)
BEGIN
    Node = Variable->Node;
    HAINSTANCE* INS = Node->Parent;
    While (INS != NULL)
        BEGIN
            HCOMPONENT* COM = INS->GetComponent();
            Switch(COM)
                BEGIN
                    Case HCREFPLANE:
                        COM->Regeneration();
                    Case HCSKETCH:
                        COM->Regeneration();
                    Case HCPART:
                        COM->Rebuild();
                END
            INS->INS->Next();
        END
    END
END
```

通过以上分析,布局设计不仅有效地表达了概念设计阶段的设计意图及维护了详细设计阶段的设计历史,完全达到了产品的自顶向下设计。

4.6 自顶向下零件设计

自底向上的产品设计是先完成零件的设计,再将零件进行装配实现产品的功能设计。而对于像皮带等非刚性的零件,它在产品中承担一定功能后的形状与原始零件状态的形状不相同,因此有此类零件的产品无法采用自底向上的方式进行设计。而自顶向下的产品设计是根据产品的功能来设计零件,具有一定功能的非刚性零件的形状是可以唯一确定的,所以这类零件只能采用自顶向下的设计方法。本文采用的布局方法就是用布局草图表示含有非刚性零件的产品功能的概念设计,布局草图通过布局设计控制非刚性零件的详细设计,从而实现非刚性的零件和刚性体配合的产品设计。例如皮带和皮带轮的装配体设计,根据产品功能的要求,需要皮带同六个皮带轮配合从而进行传动,如图 4.10 所示。

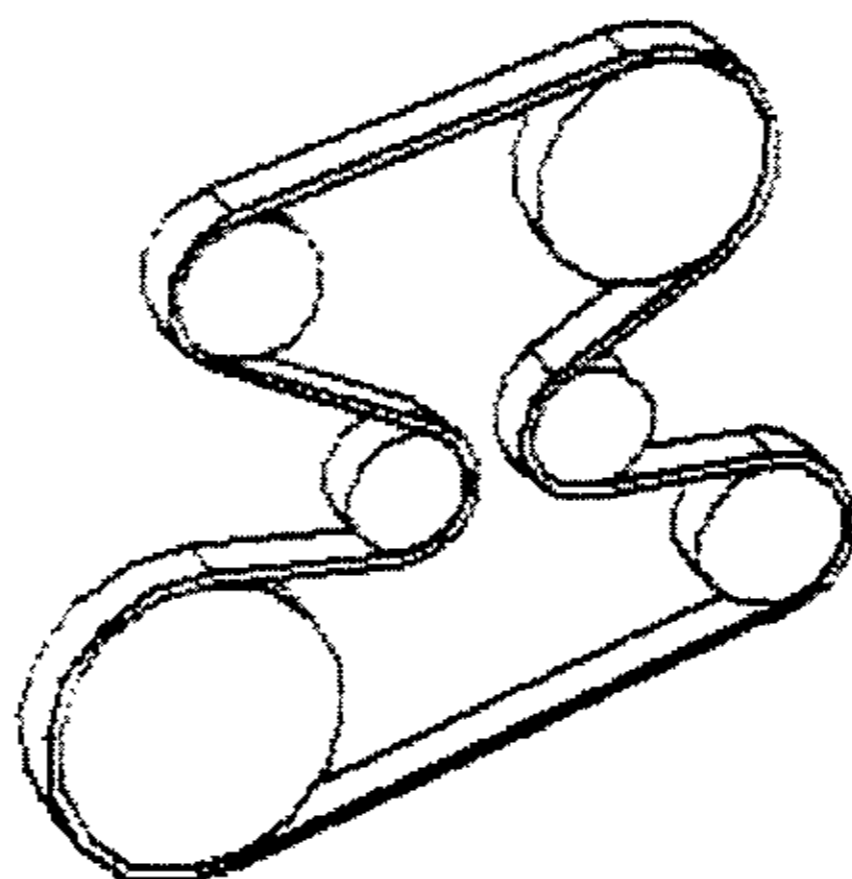


图 4.10 带类零件装配体

为了实现此产品的设计,需用装配环境的布局草图表达产品的功能,布局草图如图 4.11 所示,再用布局设计进行产品设计。其操作步骤为:

- 1) 对布局草图元素进行编码;
- 2) 进行布局设计,建立布局、布局零件和布局草图的历史编码;
- 3) 布局历史编码和草图元素编码向零件环境进行映射;
- 4) 利用零件环境的草图元素进行零件设计;

5) 对编码进行解码从而构造此装配体。

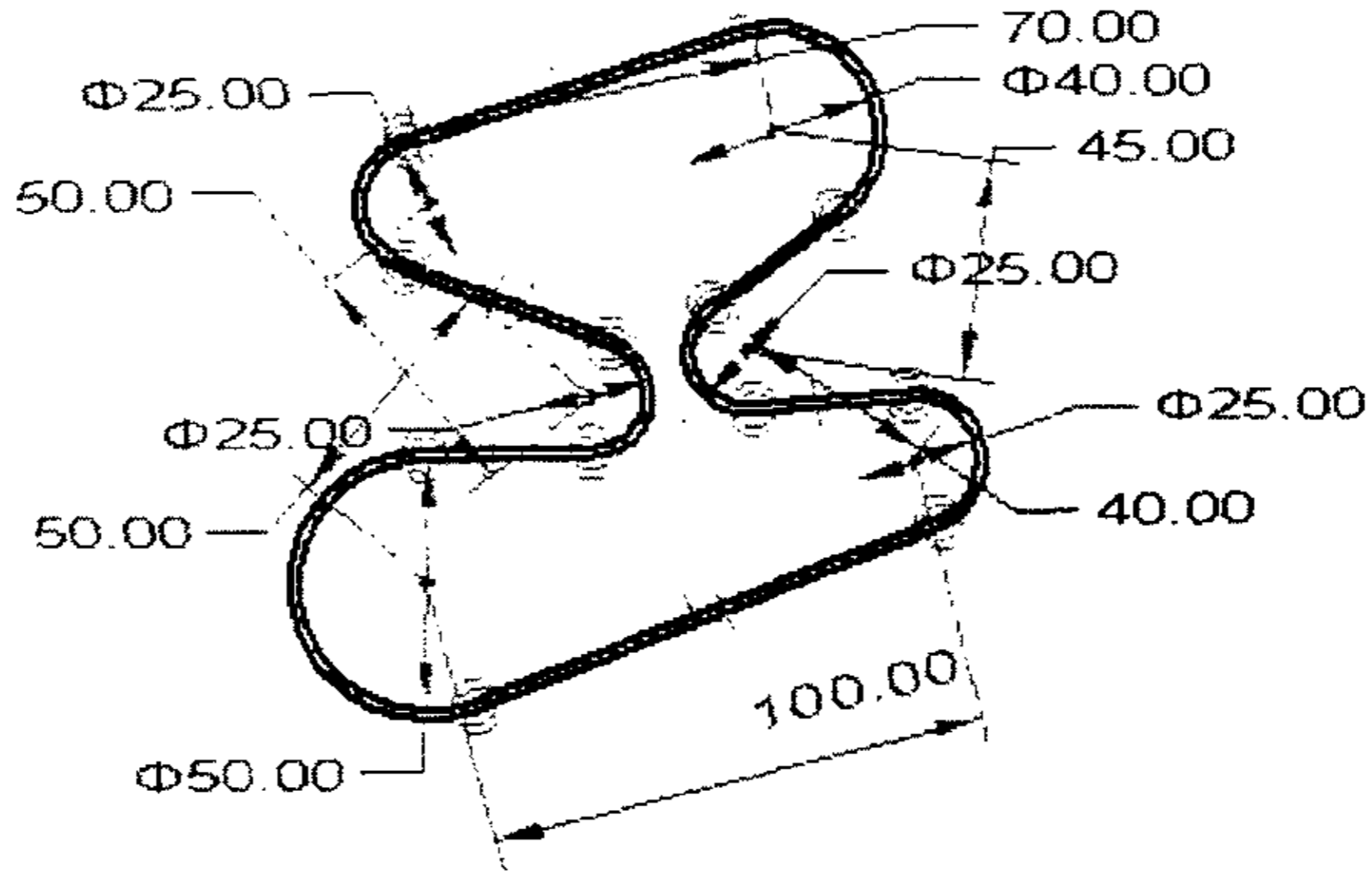


图 4.11 带类装配体布局草图

当发现两轮心的距离尺寸 40mm 需要调整时,只需要修改布局草图中的距离尺寸变量值,变量修改后的布局草图如图 4.12 所示。

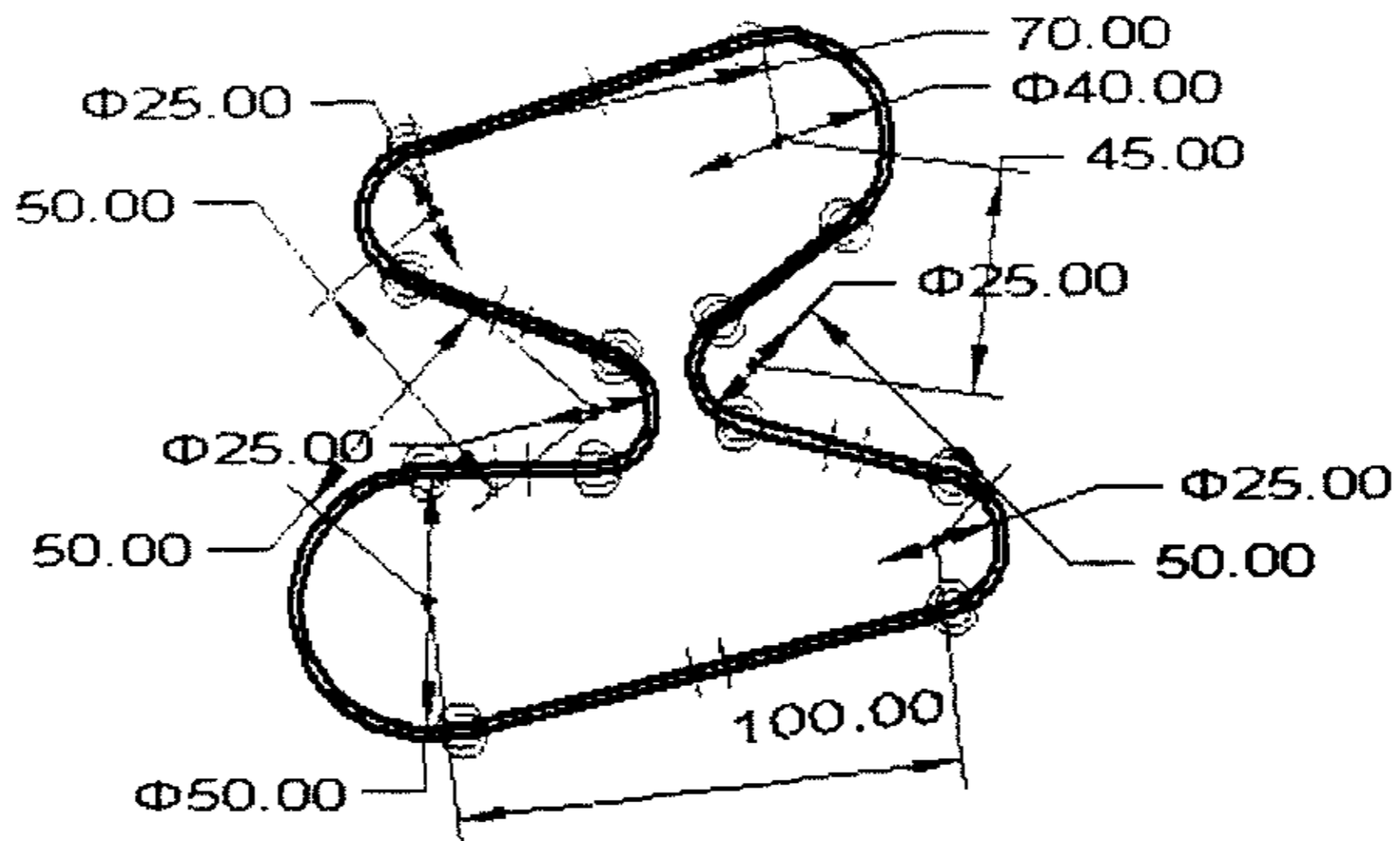


图 4.12 变量修改后的布局草图

当布局草图变量发生变化后,产品的设计会根据布局草图变量的修改而对布局零

件进行重构。重构后的带类装配体如图 4.13 所示。

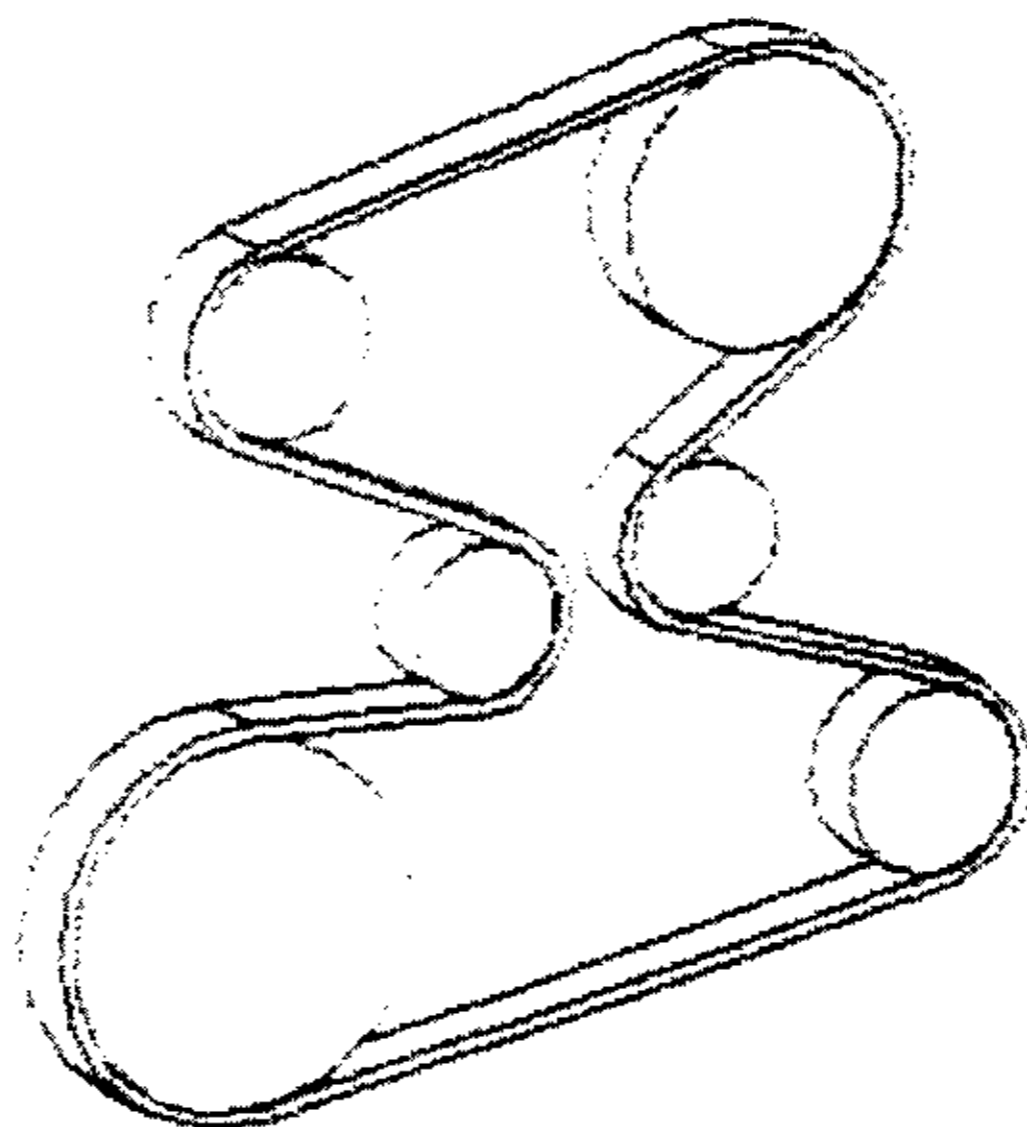


图 4.13 变量改变后的带类装配体

4.7 本章小结

布局设计的关键是如何维护布局设计的历史过程,本文针对布局历史设计过程的维护,提出了基于历史的布局基本框架,在三维设计系统中给出了维护布局设计历史的数据模型。并在布局对象的编码/解码的原理基础上,提出了布局编码的映射机制,实现了布局草图对零件详细设计过程的控制。最后以自底向上的设计方法无法实现的带类装配体设计为例,说明了布局设计历史过程的维护。

5 系统简介与实例应用

【本章摘要】 本文设计和开发了华中科技大学 CAD 中心和天喻信息软件公司的三维变量设计系统天喻三维 CAD 中的布局模块。本章主要介绍该模块的特点、使用方法, 并给出了相关的实例。

5.1 系统概述

天喻 CAD 三维版是面向机械设计领域而设计开发的二、三维一体化产品造型和设计平台。具有强大的参数化特征造型、曲面造型功能和二维视图的自动生成能力, 结合完整的二维图形系统, 使用户不仅能得到精美的三维造型实体, 而且能绘出满足实际要求的工程图。使广大工程师能够在熟悉的二维工程绘图空间和更为强大的三维设计空间自由驰骋。它的主要特点是:

1) Windows 界面风格使学习使用更为方便

全中文 Windows 界面, 支持 Window NT/98 操作系统, 使学习使用更为方便。

2) 二维和三维完美结合

在天喻 CAD 中包含完整的二维图形系统, 二维的轮廓数据可以直接输入到三维的草图中, 方便地利用已有的二维工程图中提取有用的轮廓来生成三维的特征, 加快了零件造型的过程。通过视图的自动生成功能可以快速生成标准视图和剖视图, 从三维零件模型快速得到视图数据, 然后对视图进行修改、标注等操作, 生成真正意义上的完备的工程图纸。

3) 草图设计独具匠心

独一无二的草图可重用特性使零件设计更加方便、快捷。除了提供丰富多样的绘图工具、编辑工具和尺寸标注功能外, 还提供了绘图和编辑过程中的所见即所得功能, 使轮廓生成快速、准确。

4) 全模型的动态导航和关联能力

全模型的动态导航和关联能力可建立与其他草图和当前零件体上的元素的约束关系, 这种关系在模型变更后依然可以保持。

5) 可视化的约束管理

草图中的所有约束都是可视化的, 在草图的创建和编辑过程中对草图的约束状态一目了然。

6) 草图诊断医生

草图诊断医生可以对草图进行分析并定位开环、自相交和共线元素，帮助用户快速对草图进行修正。

7) 特征造型直观轻松

在提供拉伸、旋转、放样、扫掠、倒角、圆角、切角、阵列、镜像和参考特征的同时，又提供了独特的孔特征、自适应的肋板、参数化的抽壳功能及曲面与实体融合的特性，使复杂形体的创建变得非常方便、容易、快捷。许多类型的特征和曲面在创建和编辑定义时的预览功能使特征创建过程更加直观。

8) 无限级的 Undo/Redo 能力

无限级的 Undo/Redo 能力使设计过程更轻松。零件级和草图级的分层设计可以从设计过程中的一个任意节点 Undo/Redo 到另外的一个任意节点。

9) 灵活的变量表功能

在变量表中包含丰富的函数类型，可以定义用户自定义变量，给变量添加表达式，修改变量值，并对零件进行驱动。变量表中的变量会随着对模型的 Undo/Redo 自动进行 Undo/Redo。

10) 自顶向下装配设计的支持

在装配环境下进行产品的设计，用二维布局草图体现概念设计，控制三维零件的详细设计。

11) 高级渲染效果逼真

对所生成的零件进行高级渲染得到各种具有真实感效果的图形。可提供多种类型的光源并对光强、颜色、位置、衰减、阴影等参数进行调节。对材质设置中的多种纹理以及效果设置参数的调节可使渲染更加直观真实，满足个性化的需要。

12) 数据接口丰富多样

在丰富多样的数据接口中，二维数据接口包括 DWG 和 DXF，三维数据接口包括 SAT、STL、IGES、VDA 和 VRML(1.0 和 2.0)。这些接口使得与其它流行的 CAD 系统的数据交换畅通自如，虚拟产品的开发与设计成为现实。

13) 良好的开放性

系统作为开放的平台，用户可在此基础上开发各种应用系统。

5.2 布局简介及实例

5.2.1 布局简介

为了支持概念设计、实现概念设计对详细设计的控制功能，布局必须具备以下相

关功能：(1) 对布局参考平面的操作；(2)对布局草图的操作；(3) 对布局设计的草图；(4)用布局控制零件的详细操作。如图 5.1 是包含有布局功能的三维变量设计系统 intesolid2.0 的界面。左边为装配管理树；他显示一个产品从概念设计到详细设计的历史过程，右边则是设计区，进行布局设计时的所有工作都会在此区域显示出来。

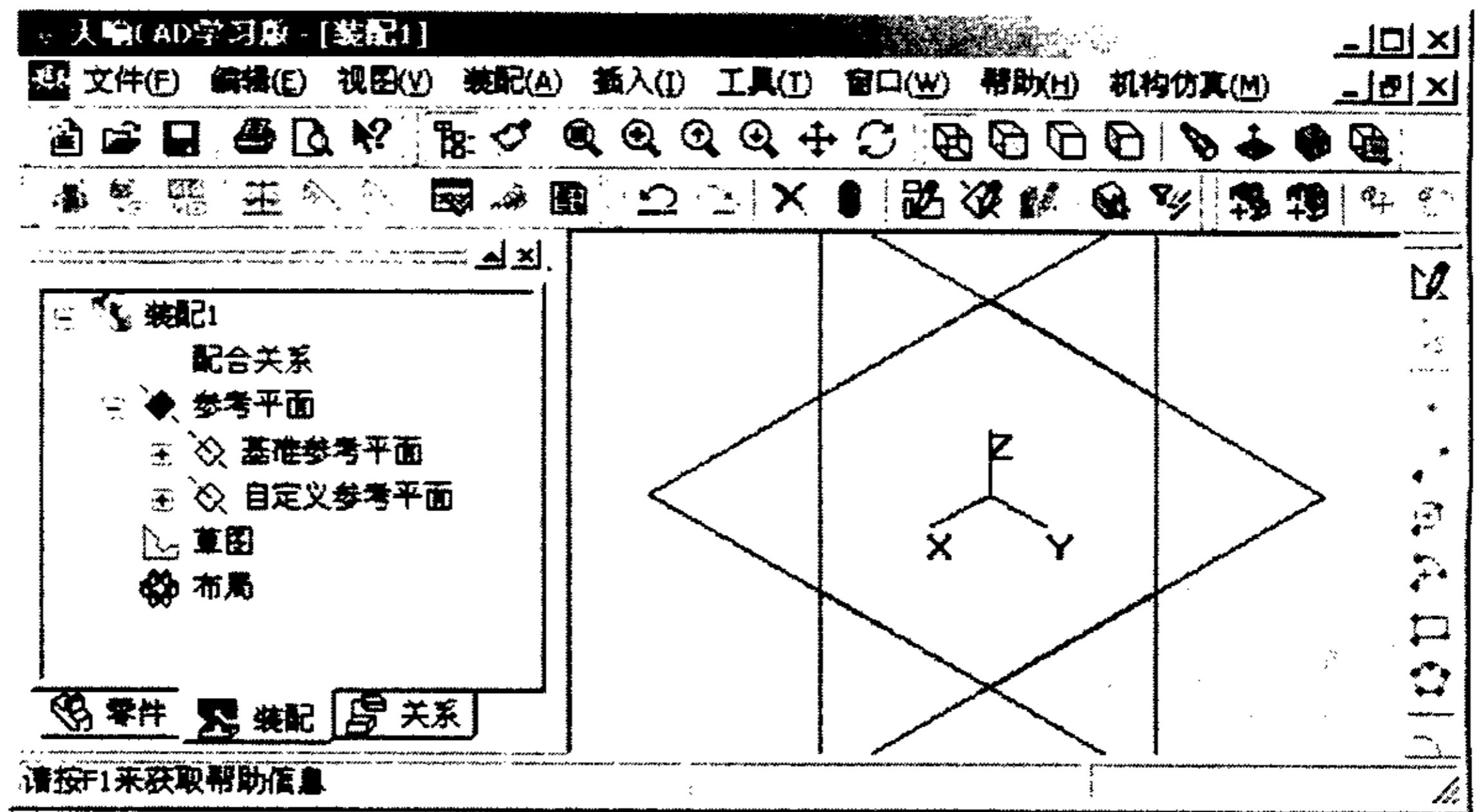


图 5.1 实现布局设计的系统界面

在系统的左边我们可以看到有三种树状结构，零件、装配和关系，零件布局设计中各布局零件和它们的相互关系；装配显示整个产品设计全历史过程；关系则显示构成产品功能的各种配合关系。装配 1 是对所设计的装配体产品的命名，此名称可以进行自定义，例如将装配 1 改名为减速箱等。

5.2.2 布局所属对象设计简介


1) 布局参考面的设计

布局采用二维草图对三维造型的控制，所以必须有一个二维环境，在三维设计系统中，用参考平面作为二维草图设计的环境。通常三维系统都会提供三个方向的基准参考平面，但是基准参考平面远远不能满足实体造型的需要，在此引入自定义参考平面。

自定义参考平面是在基准参考面或者它上一级草图元素的基础上创建的，包括偏移面、倾斜面、三点定面、过点与平面平行、过点与直线、过边端点与边垂直、中面、

两点定面。

现在对 intesolid2.0 系统中布局参考面的操作作详细讲解。

单击图标 ，或者在插入工具栏选择插入参考平面一项，向系统中发出插入参考平面的命令，弹出如图 5.2 所示参考平面对话框：

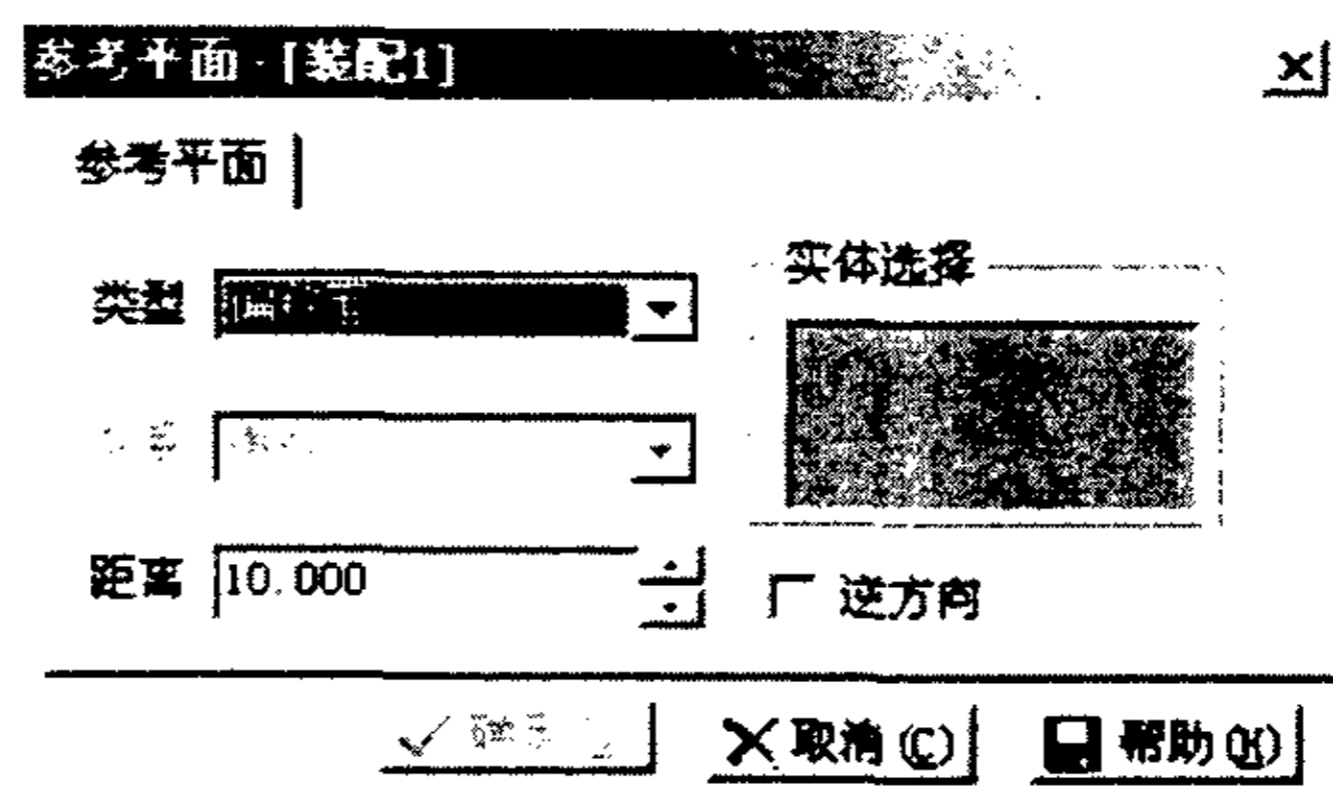


图 5.2 参考平面

在此对话框参考平面类型一栏系统默认的自定义类型为偏移面，对其它类型的选择可以根据下拉菜单对参考面的类型进行选择。参考面的类型不同则构造次参考平面的实体不同，因此在实体选择列举框中实体的选择需要根据参考平面的类型来决定。例如，偏移面只是相对于某一个平面偏移，一次只需要选择一个参考面作为实体，而三点定面则需要选择三个草图元素点。对话框中的距离编辑项为距离和角度编辑项的合成，当参考平面为偏移面，需要添加的为距离关系尺寸，如果是倾斜面，添加的应该是倾斜角度尺寸，此时对话框显示的为角度编辑项。所以此处显示距离或角度也需要参考构造的参考平面类型。实体选择下面的逆方向复选框则构造参考平面的法矢方向，系统缺省构造参考平面的法矢为正方向，如果需要方向时，只需要单击此复选框。各种参考平面构造参数选择如表 5.1 所示。

参考平面构造以后，如果需要对其重新编辑，只需要选中需要编辑的参考平面，单击右键，选择编辑定义则会对参考平面进行重新构造。参考平面的右键菜单如图 5.2 所示。因为基准参考平面为系统产生，所以不允许进行编辑定义。在右键菜单中有一项为显示，此项功能是显示或隐藏参考平面，当参考平面为隐藏状态时，单击显示，则可以显示参考平面，当显示前为对勾所标记，表明参考面为显示状态。

表 5.1 布局面构造参数表

参考平面类型	实体选择	距离或角度参数	逆方向
偏移面	已知参考平面	偏移距离	与缺省方向关系
倾斜面	参考平面 与此面平行的草图直线	倾斜角度	同上
三点定面	草图点 1 草图点 2 草图点 3		同上
过点与平面平行	草图点 已知参考平面		同上
过点与直线	草图点 草图直线		同上
过边端点与边垂直	草图曲线或直线边 草图边端点		同上
中面	参考平面 1 参考平面 2		同上
两点定面	草图点 1 草图点 2		同上

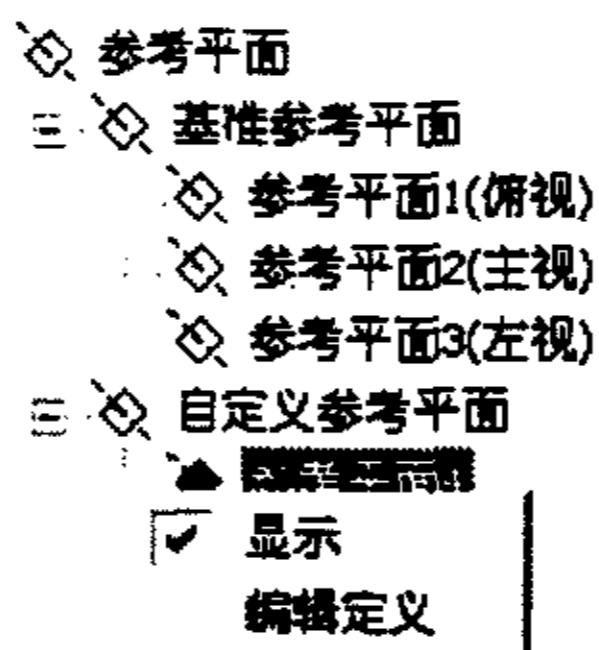




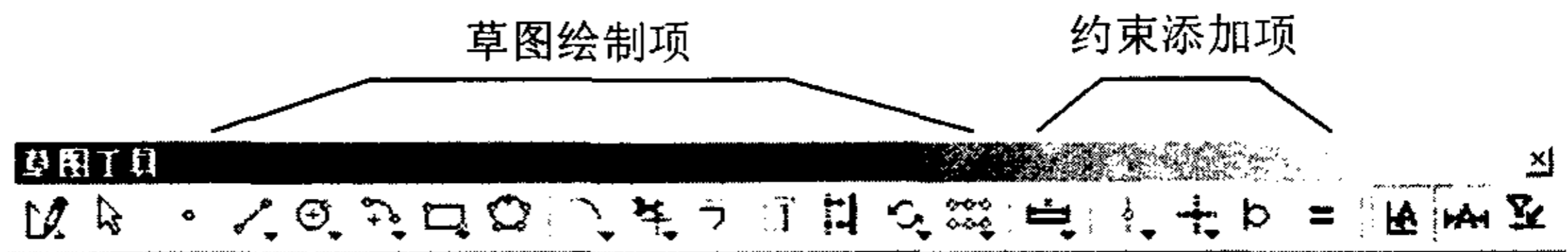
图 5.2 参考平面右键菜单

如果发现某一参考面影响布局或者其它布局对象时，需要对其进行删除操作，选中需要删除的参考平面，单击图标  或者选择编辑工具栏中的删除命令对参考面进

行删除操作。当参考面删除后，管理树和装配文档自动进行更新，删除此参考面和此参考面上的草图，以及由此草图所构造的所有布局和布局控制的零件。

2) 草图的设计

布局是用二维的草图信息来表示产品功能的概念设计，因此布局草图的设计在布局设计充当非常重要的角色。在本系统中，选择一个参考平面后，单击图标或者选择插入工具栏中草图向草图工程发出插入草图的命令，草图工程提供的草图绘制工具条使用户更方便的进行草图插入等编辑工作，草图工具条如图 5.3 所示。



5.3 草图工具条




选择草图绘制项对草图元素进行设计，选择约束添加项为草图元素的约束关系进行定义。当草图操作完成后，只需单击则可以关闭草图，还可以用草图右键菜单对草图关闭进行操作，草图右键菜单如图 5.4 所示。



图 5.4 草图关闭右键菜单

选择右键菜单中的关闭草图同样可以关闭草图。如果草图关闭后需要再次编辑，此时单击图标可以从新打开草图构造文档，进入草图编辑状态。方便做法还可以用草图右键菜单中的编辑草图进入草图编辑状况。编辑草图右键菜单如图 5.5 所示。右键菜单中另一项隐藏草图是对草图线进行隐藏和显示，隐藏主要是防止不同草图的草图线互相之间的干扰，显示草图则是对草图元素引用或者参考需要。


当草图处于关闭状态，需要删除这个草图时单击图标或者选择编辑工具栏中

的删除命令对草图进行删除操作，草图删除后，管理树和装配文档自动进行更新，删除引用草图元素构造的参考面和它上的一起元素以及以此草图而构造的布局设计。当草图处于编辑状态，所作的草图元素的删除同样会反应到装配文档中布局的生成。



图 5.5 草图编辑右键菜单

3) 布局设计

布局设计主要是实现装配环境的布局草图向零件环境进行映射。因此在布局草图处于关闭状态，选择装配工具条中的布局设计命令，实现布局草图向零件环境的映射。此映射到零件环境的布局草图将对零件详细设计进行控制。当需要对布局进行重新编辑时，单击布局右键菜单 **编辑布局** 即可实现布局草图重新向零件环境进行映射的操作。如果发现布局设计不合理需要删除的时候，选中需要删除的布局，单击图标  或者选择编辑工具栏中的删除命令对布局进行删除操作。布局删除后，管理树和装配文档自动进行更新，删除布局所对应的零件和此布局。

4) 布局零件设计

当布局设计将布局草图信息映射到零件环境后，此后则用映射的布局草图信息对布局零件进行详细设计。此设计过程不在本文叙述，本文关键简介装配环境对布局零件的控制作用。布局零件设计完成以后，装配文档管理布局零件的一些操作，布局零件的操作项可以用布局零件的右键菜单来描述，布局零件右键菜单如图 5.6 所示。

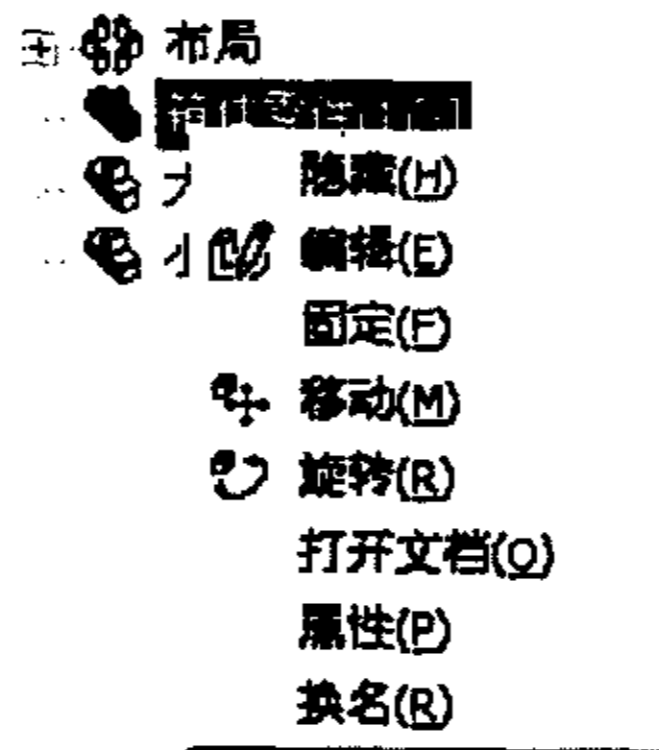


图 5.6 布局零件右键菜单

5.2.3 布局实例

本文以减速箱的自顶向下构造过程为例来说明布局是支持概念设计的产品设计。

减速箱由下箱体、大齿轮轴、小齿轮轴三个关键零件配合而成。在设计中箱体大小根据两齿轮轴的中心距大小来确定，因此齿轮箱体空腔和外轮廓大小都由齿轮轴中心线的距离来控制。按照这个功能要求可以给出表达减速箱设计关系的布局草图如图 5.7 所示。

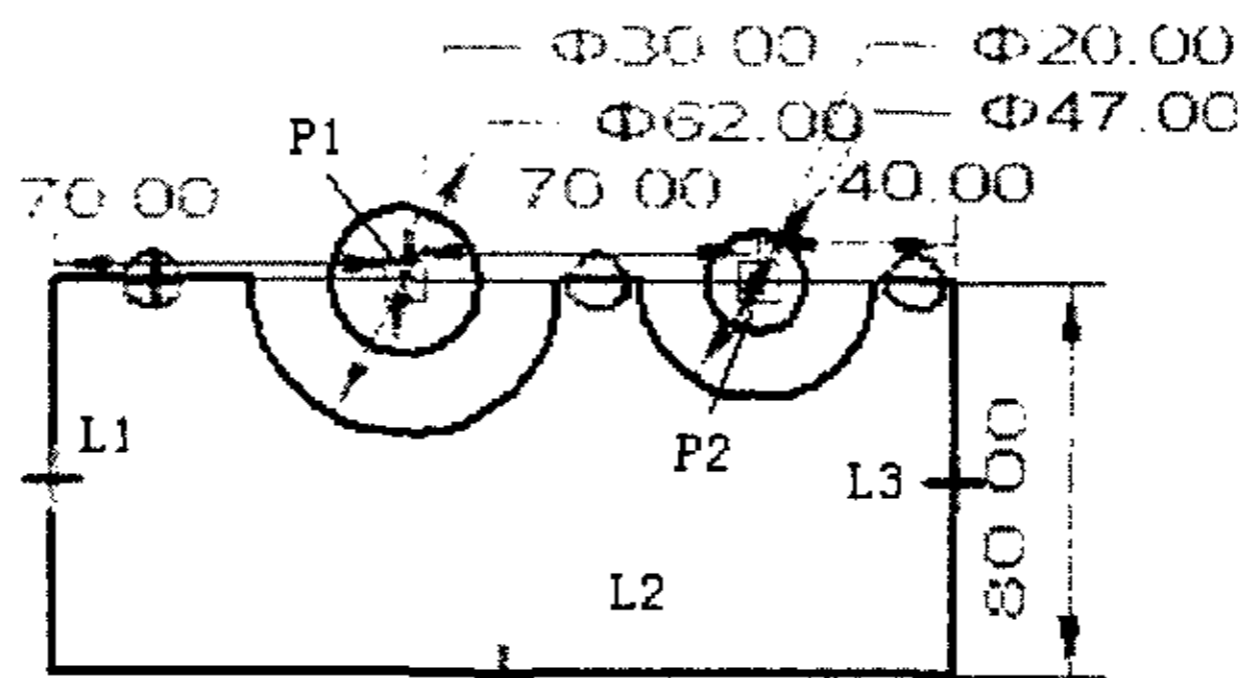


图 5.7 减速箱布局草图

由图 5.7 的减速箱布局草图可知，草图元素的属性尺寸控制各零件的大小，例如尺寸 $\phi 30.00$ 和 $\phi 20.00$ 分别控制大小齿轮轴轴承配合处的轴径；P1 和 P2 的距离尺寸 80mm 为两齿轮轴的中心距。以零件的构造原理，下箱体为一个零件，大小齿轮轴为两个零件，因此此布局草图应该表达此三个零件的配合关系以及简单的线轮廓。在草图中，齿轮轴的简单形状与箱体轴承孔为同心圆，以此保证齿轮轴与轴承孔之间的同轴线配合关系，下箱体侧壁 L3 与齿轮轴圆心 P2 距离为 40mm 控制箱体尺寸随着齿轮轴中心距的变化而变化，两齿轮轴布局草图的圆心 P1, P2 同在同一条直线上，以此来约束两齿轮轴同高，防止齿轮啮合传动时的抖动。草图元素和约束关系构造完后，就可以退出草图进行详细的零件设计。

根据布局草图对布局零件进行编辑，编辑的下箱体零件如图 5.8 所示。

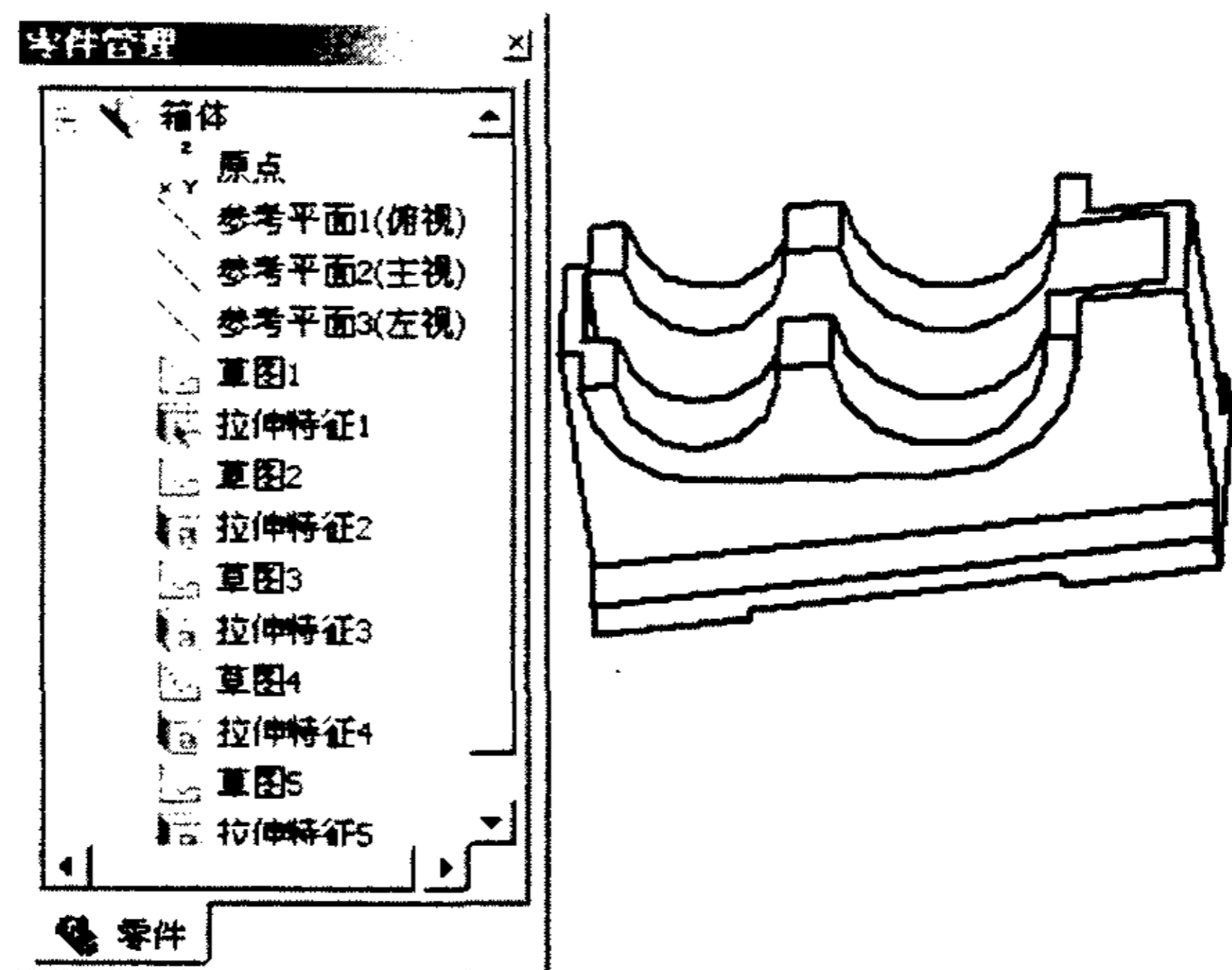


图 5.8 箱体零件设计过程

保存下箱体零件然后关闭零件文档，此下箱体零件则会显示在装配编辑文档中，同时在装配树上显示它的零件名。同理可以设计大小齿轮轴零件如图 5.9 所示。

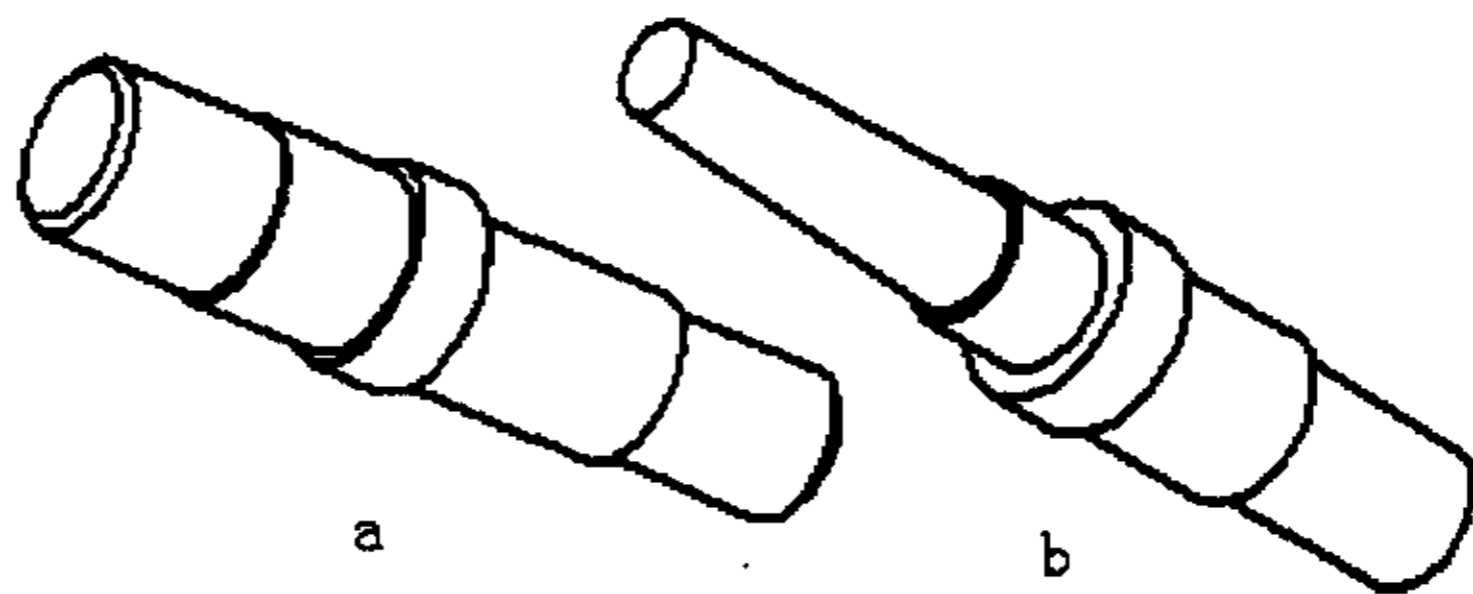


图 5.9 大小齿轮轴

完成零件设计后，布局设计的减速箱显示在装配环境中，并且装配历史树上记录了减速箱设计的历史过程。如图 5.10 所示。

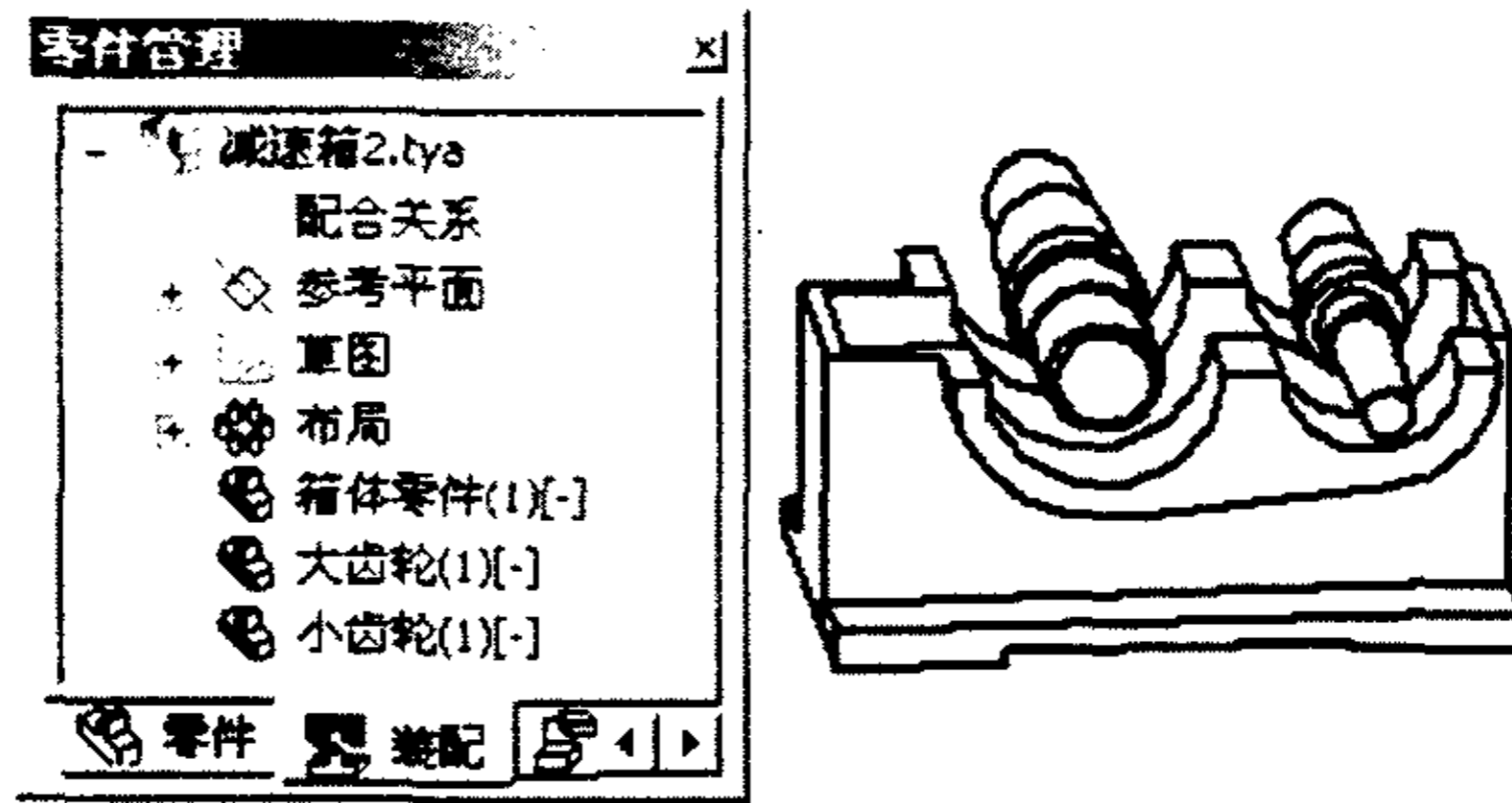


图 5.10 减速箱装配体

在减速箱的设计中，箱体尺寸必须随着中心轴线的距离改变而改变，当调整两齿轮轴中心距时，即将 70mm 改成 80mm 时，减速箱的布局草图会发生如图 5.11 的尺寸变更。

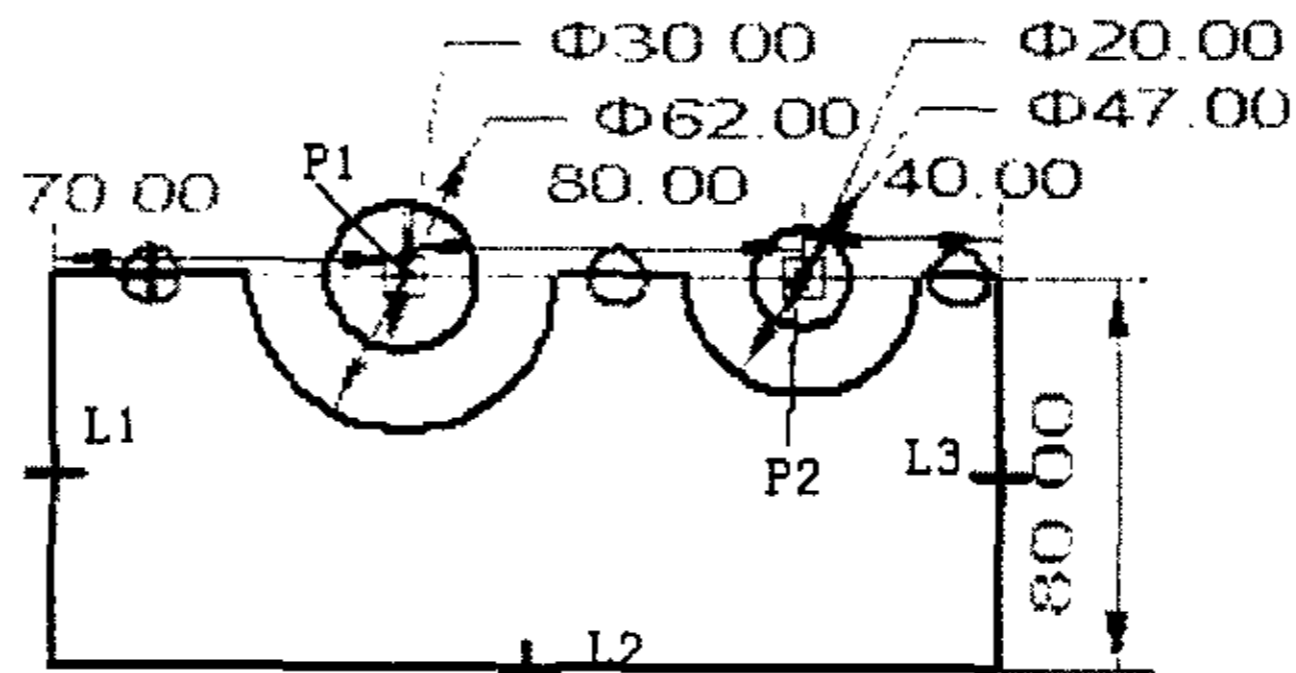


图 5.11 布局草图尺寸变更

关闭草图，减速箱所有布局零件的尺寸会布局草图尺寸的变更而自动进行尺寸更新，布局草图变量更改导致减速箱变体如图 5.12 所示。

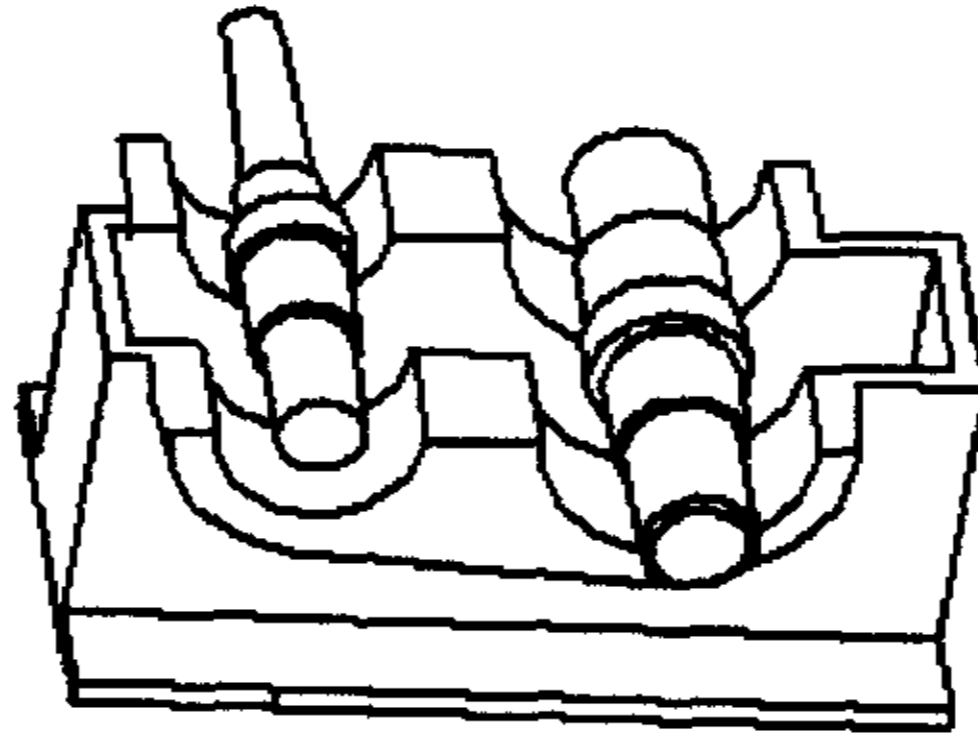


图 5.12 中心距变化后的减速箱

由上例可知，布局设计方法使产品设计的局部修改变得快速而简便。

6 结论与展望

6.1 全文总结

目前,支持产品的概念设计已经成为三维 CAD 系统研究的热点。本文针对这类问题的相关理论进行了深入地研究用布局的方法实现对产品概念设计的支持使三维 CAD 系统更能够符合工程设计的需要,是产品设计过程变得简单、快捷;将工程中常见的三维闭环问题转化为二维闭环约束问题进行求解,减少闭环约束求解的不稳定性,结合华中科技大学天喻软件公司和华中科技大学 CAD 中心研究开发的三维变量设计造型软件天喻三维 CAD 取得了一些成果,归纳如下:

- 1) 从三维变量设计系统、产品布局设计两个方面,总结了国内外相关的研究成果和最新研究动态,分析了当前三维变量设计系统处理变量化技术中的三维闭环约束问题算法的基础上,结合装配的层次模型,提出了基于三维变量化系统的布局模型,将工程中常见的三维闭环约束问题转化为二维闭环约束问题,从而为解决此类问题求解提供了可行的方案。。
- 2) 对实现三维设计系统布局所需要的各种关系进行研究。给出布局面、布局草图等定义,并且研究它们需要遵循的层次关系,同时在系统中给出实现这种层次关系所用的数据结构和基本算法。
- 3) 本文结合基于历史的编码/解码的原理,提出了基于编码/解码的映射机制,实现了布局设计基于历史的重构,达到了概念设计中设计意图更改而导致整个设计过程随之变更的要求,同时解决了自顶向下的零件造型问题。。
- 4) 用布局的方法实现了自顶向下的产品设计,并且使工程中常见的三维闭环问题的求解变得简明而有效。布局设计在三维软件天喻 CAD 系统中得以实现,增强了系统的功能。

6.2 进一步的研究工作

布局的研究在三维变量化设计系统中是一个具有重要意义的难题,其核心在于如何实现布局草图对零件详细设计的控制。本文在布局研究方面的重点在于怎样将基于历史的装配环境的布局关系映射到零件环境对零件设计进行控制,采用了基于历史的统一的编码/解码原理,定义了实现支持产品概念设计的各种布局关系,并用映射的

机制将这些关系映射到零件环境进行对零件操作的控制。但要做到用布局完全体现产品概念设计还有许多工作要做：

- 1) 实际设计中，分解的产品功能都存在一定的关系，如何将这种关系用布局之间的约束关系进行表达需要进行进一步研究。
- 2) 布局能使设计者对产品概念设计方案进行快速而有效地修改，但是在产品概念设计方案的设计显得不力。本文认为在这方面需要进行大量研究。
- 3) 布局的方法可以解决工程中常见的三维闭环约束问题，但是对于所有的三维闭环约束问题的求解还需更多的尝试。

致 谢

本文的研究工作是在尹文生、王启付副教授和黄运保博士的悉心指导下完成的。三年来，本人在思想、学业、生活诸多方面得到导师的精心指导和关怀。导师们精深广博的知识，敏锐的分析洞察力使作者受益匪浅；他们那严谨治学的态度，精益求精的科学精神，脚踏实地的工作作风为作者树立了终身的榜样；导师们学习上呕心沥血的指导，生活中无微不至的关怀，使作者深深感动，并将永远激励学生去努力攀登科学和事业的高峰。值此论文完成之际，谨向恩师致以最诚挚的感谢和崇高的敬意！

深深感谢王启付老师、朱林老师和陈立平老师。在本人三年的研究生期间，老师们在课程学习和课题研究方面都给予本人很多建设性的建议和指导，并在生活上给予本人悉心的关心和帮助，在此表示深深的感谢。

在本文研究过程中，得到了岳建鹏硕士、朱文玄硕士、王波兴博士、熊传光硕士及 CAD 事业部 Solid 小组众位的支持和帮助，在与他们的共事交流中，我深受启发，促成了本文思想的形成与成熟，是他们的无私合作促成了本文研究的顺利进行，在此表示真诚的感谢。

感谢刘继红老师、肖人彬老师、罗宏志老师以及 CAD 中心全体老师和同学们给予的帮助和关心。

感谢同窗好友吴磊、乔雨等以及机械学院 99 级全体硕士生同学和向东生、程才、蒋勇、夏鸿建、柯敏等同学，正是他们给了作者学习和生活上的莫大帮助，使作者三年的学习生活充满了欢笑与阳光。

深深地感谢亲爱的父母亲的养育之恩和二十多年来在作者求学道路上无私奉献的精神，父母的关怀和厚望永远激励本人不断地进取！

把由衷的感激送给亲爱的舅舅、舅妈和奶奶以及其他亲人，是他们的理解、支持和帮助使得我的学业能够顺利完成，同时他们的期望和嘱托也将永远鞭策着我向前！

感谢所有关心和帮助过我的人们！感谢所有爱我的人们！

美丽的华工园为我提供了优美的学习和工作环境，教给我赖以生存的知识，指引我前进的方向，让我深深留恋。

李海艳
2002年4月

参考文献

- [1] 源清, 肖文. CAD 技术发展历程概述. 计算机辅助设计与制造. 1998,1:3~6
- [2] Sohlenius G .Concurrent engineering, Annals of the CIRP.1992,41(2):645~655
- [3] 源清, 肖文. 简析九十年代主流 CAD 造型基础技术. 计算机辅助设计与制造. 1998,2:1~4
- [4] 孙家广等. 计算机辅助设计技术基础. 北京: 清华大学出版社. 1990
- [5] Thornton A C, Johnson A. Constraint specification and satisfaction in embodiment design. Proc. Int. Conf. on Engineering Design. ICED 93, the Hague. 1993, 1319~1326
- [6] Thornton A C. Constraint specification and satisfaction in embodiment design. PhD dissertation, University of Cambridge. 1993
- [7] Ullman D G, Dietterich T G, Stauffer L. A model of the mechanical design process based on empirical data. Artificial Intelligence for Engineering, Design, Analysis and Manufacturing. 1988, (2): 33~52
- [8] Kramer G A. A solving geometric constraint systems, a case study in kinematics. The MIT Press, Cambridge, Massachusetts, 1992
- [9] 朱志良. 新一代三维 CAD 系统——SOLID EDGE. 计算机辅助设计与制造. 1999, 2: 36~39
- [10] 庞士宗等编著. SOLID EDGE 七日通. 北京: 机械工业出版社. 1999,4
- [11] SolidWorks 公司, SolidWorks 用户指南
- [12] Light R and Gossard. Modification of geometric models through variational geometry. Computer-Aided Design. 1982,14:209~214
- [13] J U .Turner. Relative position of parts in assemblies using mathematical programming . Computer Aided Design. Volume 22 No.7 September 1990:394~398
- [14]K.W. Lee & G. Andrews. Inference of the position of components in an assembly: part 2 . Computer Aided Design. Volume 17 No.1 Jan/Feb. 1985:20~24
- [15]David Neville Recheleau and KunwooLee , "System for interactive assembly modeling". Computer-Aided Design. 1987, 19(2):65~71
- [16]Rajneet Sodhi and J U . Turner. Relative positioning of variational part models for design analysis . Computer Aided Design. Volume 26 No.5 May 1994:366~377
- [17]康友树. 变量装配设计原理及方法研究. 华中理工大学博士学位论文. 1996
- [18]尹文生. 支持自顶向下的装配建模研究. 华中理工大学博士学位论文. 1998
- [19]吴永明. 三维几何约束管理的理论和方法研究. 华中理工大学博士学位论文. 1998
- [20]Kramer G A. A Geometric Constraint Engine. Artificial Intelligence.1992 ,58

- [21]陈立平等. 一种新的面向参数化绘图的约束管理技术. 软件学报.1996, 7(7)
- [22]陈立平.几何约束系统最大归约理论及应用研究. 华中理工大学博士学位论文. 1995
- [23]Bouma W, et al. Geometric Constraint Solver. CAD.1995, 27(6)
- [24]Ioannis Fudos. Editable Representations for 2 D Geometric Design. Master thesis of Purdue University. 1993
- [25]Markov Lev A, Fox Jeffrey R, Blank John H. Optimization Techniques for Two-dimensional Placement. In:DesignAutomation Conference(21st),Albuquerque N. M. 1984,652~654
- [26]邱英汉. 二维实体布局中实体初始合理摆放位置的研究. 计算机工程. 1997, 23(5): 48~51
- [27]戴佐, 查建中, 袁俊良. 三维实体布局中干涉检验的八叉树方法. 计算机辅助设计与图形学学报. 1995,7(4): 257~262
- [28]段国林, 查建中, 林建平等. 模拟退火法在钟手表机芯布局中的应用. 计算机辅助设计与图形学学报. 1999, 11(3):276~279
- [29]Sweeney P E, Patcmoster E R. Cutting and Packing Problems: A Categorized Application- Oriented Research, Bibliography. J. Opt. Res. Soc. 1 992 ,43(7):691~706
- [30]Coyne R D, Postmus A G. Spatial applications of neural networks in computer- aided design. Artificial Intelligencein Engineering.1990 ,5(1):9~22
- [31]吴慧中, 王英林. 一种立体空间布局模型及布局算法. 计算机学报. 1994, 17(11): 835~841
- [32]王英林, 吴慧中. 空间布局的约束图方法. 软件学报. 1998, 9 (3), 200~204
- [33]Montreuil B. Requirements for Representation of Domain Knowledge in Intelligent for Layout Design. Computer-Aided Design. 1990, 22:97~108
- [34]Oxman R. Prior Knowledge in design:A dynamic knowledge- based model of design and creativity. Design Stud-ies. 1990, 11(1):17~28
- [35]A. Voß .C- H. Coulon. W Grather, B. Linowski, J. Schaaf, B Bartsch- Spori, K. Borner,E. C. Tammer, H. Durschke, and M. Knauff. Retrieval of similar layouts- about a very hybrid approach in fabel. In J. S. Gero and F. Sud-weeks, editors, The Netherlands.1994, Kluwer, Academic Publishers. Artificial Intelligence in Design'. 94: 625~640
- [36]袁苗龙, 张新访, 周济. 智能支持的汽车驾驶室内布置方案设计研究. 计算机集成制造. 1996, 4:23~26
- [37]包恩伟, 孙守迁. 基于组件特征模型的产品布局设计. 工程图学学报.1998,(3)16~19
- [38]杨勤, 罗仕鉴, 朱上上. 产品空间布局设计研究. 贵州工业大学学报 (自然科学版) .2001,30(4):19~22
- [39]Pro/Engineer—Assembly Modeling User's Guide. Ver.12.0
- [40]Pro/Engineer—Feature-Based Modeling User's Guide. Ver.12.0
-

- [41]I-Deas Solid Modeling User's Guide-Section VI,Assembly Modeling.Ver3.0
- [42]Hillyard R C, Braid I C. Analysis of dimensions and tolerance in Computer Aided mechanical design. Computer-Aided Design. 1978, 10(3): 161~166
- [43]Gossard D, Zuffante R P, Hiroshi, Sakurai. Representing Dimension, Tolerances and Features in MCAE Systems. IEEE CG&A. 1988: 51~59
- [44]尹文生, 罗宏志, 王同洋等.支持概念设计装配模型的零件管理策略研究.华中理工大学学报.1997,25(10):1~3
- [45]姜华.机械产品的制品规划研究.华中科技大学博士学位论文.1997
- [46]尹文生, 吴明华, 周济.支持概念设计的装配模型基本数据结构研究.计算机应用研究.1997,6:178~180
- [47]陆薇, 刘强, 周琛等.基于组件的开放式 CAD 系统模型.软件学报.1998,9(9): 651~655
- [48]尹文生.基于广义环图树的装配设计模型.华中理工大学学报.1999,29(8):3~5
- [49]Lee.K and Gossard.D C "A hierachical data structer for representing assemblise:part1". Computer-Aided Design.1985, 17(1):15~19
- [50]谭建荣, 张树有, 魏修亭等.产品装配信息概念模型及自组织映射方法.中国机械工程.1998,9:66~69
- [51]Bourjault A. Methodology of assembly automation: a now approach. Proceeding of the 2nd Intenational Conference on Robotics & factoties of the future, Sandiego,California. 1097
- [52]Bourjault A and Lhote A. Modeling an assembly process. IEEE International Conference on Automation of Manufacturing Industry.1986,20(2):183~198
- [53]吴俊军.基于历史的形状特征建模理论与系统研究. 华中科技大学博士学位论文. 1998

附录 1 攻读学位期间发表论文目录

- [1] 李海艳, 尹文生. 面向三维变量设计系统的布局研究和开发. 华中科技大学学报, Vol.29, No.12, pp30-33