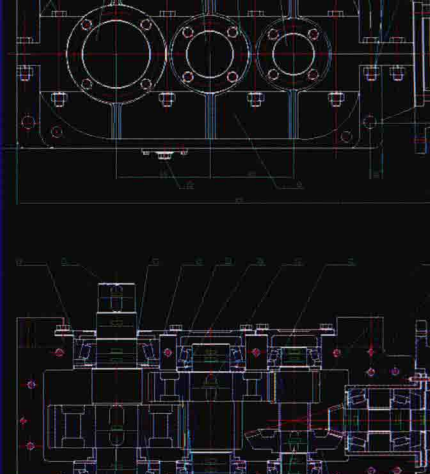


TWO DIMENSIONAL PARAMETRIC
DRAWING TECHNOLOGY OF BEARING



轴承的二维参数化 绘图技术

郭术义 著

 科学出版社

轴承的二维参数化绘图技术

郭术义 著

科学出版社

北京

内 容 简 介

本书结合常用滚动轴承的国家标准以及此类零部件的机械制图标准,利用AutoLISP、DCL关联了此类零部件内部之间参数关系,编制了基于对话框的交互式Lisp程序,实现了常用滚动轴承的参数化绘图,使得出图效率大大增加。本书着眼于AutoCAD软件的最新功能,力求做到先进性。在结构方面,针对各类滚动轴承的Lisp的绘图过程和参数化编程,单独成章,便于学习和理解。

本书可供机械工程设计、制图等工程技术人员参考,也可作为高等院校机械工程、材料工程及相关专业教学用书。

图书在版编目(CIP)数据

轴承的二维参数化绘图技术 / 郭术义著. —北京: 科学出版社, 2016
ISBN 978-7-03-050044-1

I. ①轴… II. ①郭… III. ①轴承-二维-参数化绘图 IV. ①TH133.3

中国版本图书馆 CIP 数据核字 (2016) 第 232853 号

责任编辑: 童安齐 王杰琼 / 责任校对: 王万红

责任印制: 吕春珉 / 封面设计: 耕者设计工作室

科学出版社出版

北京东黄城根北街16号

邮政编码: 100717

<http://www.sciencep.com>

三河市骏杰印刷有限公司印刷

科学出版社发行 各地新华书店经销

*

2016年11月第一版 开本: 787×1092 1/16

2016年11月第一次印刷 印张: 8

字数: 176 000

定价: 60.00元

(如有印装质量问题, 我社负责调换〈骏杰〉)

销售部电话 010-62136230 编辑部电话 010-62135397-2021

版权所有, 侵权必究

举报电话: 010-64030229; 010-64034315; 13501151303

前 言

名目种类繁多的轴承部件作为标准件在二维装配图中经常用到，而 AutoCAD 等常用二维专业绘图软件本身却并不提供其二维图。当此类零部件的类型、参数调整时，绘图员不得不重新进行图形尺寸、点坐标等的计算、修改，装配图中出错概率大增，造成资源的极大浪费。

滚子轴承和深沟球轴承在实际的机械结构设计中应用最广泛。由于此类轴承设计过程涉及了诸如内径、外径、宽度以及倒圆角等参数，机械结构改变时，轴承类型和参数改变，设计人员不得不重新计算各变量值，重新绘图，从而导致工作量增大，出错概率大增。

为尽可能减少设计图中的错误，提高机械设计效率，使机械设计人员从繁杂的绘图工作中解放出来，采取相应技术手段使得此类零部件可以实现二维参数化自动绘图则是行之有效的解决手段。

AutoLISP 语言是 AutoCAD 软件自身携带的编程语言，利用它可对 AutoCAD 软件进行二次开发。本书结合常用滚动轴承国家标准和机械制图国家标准，利用 AutoLISP、DCL 关联了此类零部件内部之间参数关系，编制了基于对话框的交互式 Lisp 程序，实现了常用轴承的参数化绘图，出图效率大增。

本书以常用圆柱滚子轴承、圆锥滚子轴承、调心滚子轴承、深沟球轴承、调心球轴承和外球面球轴承为例详细讲述它们的参数化绘图与程序编制过程，其他类型轴承的参数化绘图也可参考此书进行。

此外，LISP 文件和 DCL 文件中的英文和标点符号，如引号、分号等都是纯英文状态输入的，从文件拷贝到本书中可能出现某些格式变化，请读者仿照例子做的时候注意修改。根据行文需要，删除了 LISP 文件和 DCL 文件中部分非必需的空格与换行符，仿写编程时请注意。

本书得到了“河南省科技攻关计划项目”的支持，在此表示感谢！

由于作者水平所限，书中难免存在不足之处，恳请读者批评指正。

目 录

| | |
|-------------------------|----|
| 第一章 绪论 | 1 |
| 1.1 常用轴承参数化二维绘图背景 | 1 |
| 1.2 软件设计方法及实现步骤 | 1 |
| 第二章 圆柱滚子轴承的参数化绘图 | 3 |
| 2.1 外圈无挡边圆柱滚子轴承参数化绘图 | 4 |
| 2.1.1 外圈无挡边圆柱滚子轴承参数 | 4 |
| 2.1.2 程序流程 | 4 |
| 2.1.3 程序说明 | 5 |
| 2.1.4 程序调试及参数化图形绘制 | 14 |
| 2.2 内圈无挡圈边柱滚子轴承参数化绘图 | 15 |
| 2.2.1 内圈无挡边圆柱滚子轴承参数 | 15 |
| 2.2.2 内圈无挡边圆柱滚子轴承程序设计思路 | 15 |
| 2.2.3 图形绘制 | 16 |
| 2.3 内圈单挡边圆柱滚子轴承参数化绘图 | 17 |
| 2.3.1 内圈单挡边圆柱滚子轴承参数 | 17 |
| 2.3.2 内圈单挡边圆柱滚子轴承设计思路 | 17 |
| 2.3.3 图形绘制 | 18 |
| 2.4 外圈单挡边圆柱滚子轴承参数化绘图 | 19 |
| 2.4.1 外圈单挡边圆柱滚子轴承参数 | 19 |
| 2.4.2 外圈单挡边圆柱滚子轴承设计思路 | 20 |
| 2.4.3 图形绘制 | 21 |
| 2.5 双列圆柱滚子轴承参数化绘图 | 21 |
| 2.5.1 双列圆柱滚子轴承参数 | 21 |
| 2.5.2 双列圆柱滚子轴承绘图程序说明 | 22 |
| 2.5.3 图形绘制 | 23 |
| 2.6 基于 DCL 的圆柱滚子轴承参数化绘图 | 24 |
| 2.6.1 DCL 界面设计 | 24 |
| 2.6.2 DCL 框架程序 | 26 |
| 2.6.3 DCL 驱动程序 | 27 |
| 2.6.4 轴承绘制 | 30 |
| 第三章 圆锥滚子轴承的参数化绘图 | 32 |
| 3.1 单列圆锥滚子轴承的参数化绘图 | 32 |
| 3.1.1 单列圆锥滚子图形及参数分析 | 32 |
| 3.1.2 绘制单列圆锥滚子轴承思路分析 | 32 |

| | | |
|------------|---------------------|-----------|
| 3.1.3 | 单列圆锥滚子轴承程序加载及图形绘制 | 36 |
| 3.2 | 双内圈双列圆锥滚子轴承参数化编程 | 37 |
| 3.2.1 | 双内圈双列圆锥滚子轴承图形及参数 | 37 |
| 3.2.2 | 程序编制思路分析 | 38 |
| 3.2.3 | 程序加载及图形绘制 | 38 |
| 3.3 | 双外圈双列圆锥滚子轴承参数化编程 | 39 |
| 3.3.1 | 双外圈双列圆锥滚子轴承图形及参数 | 39 |
| 3.3.2 | 程序编制思路分析 | 39 |
| 3.3.3 | 程序加载及图形绘制 | 41 |
| 3.4 | 圆锥滚子轴承对话框编程 | 41 |
| 3.4.1 | DCL 编程思路 | 41 |
| 3.4.2 | DCL 文件编制 | 41 |
| 3.4.3 | 幻灯片制作 | 43 |
| 3.4.4 | 对话框驱动程序设计 | 44 |
| 3.4.5 | 对话框驱动程序 | 45 |
| 3.4.6 | 对话框程序运行 | 47 |
| 第四章 | 深沟球轴承的参数化绘图 | 48 |
| 4.1 | 深沟球对话框程序编写 | 48 |
| 4.1.1 | 深沟球轴承基本参数 | 48 |
| 4.1.2 | 深沟球轴承对话框界面 | 48 |
| 4.1.3 | 深沟球轴承对话框程序 | 49 |
| 4.1.4 | 深沟球轴承幻灯片制作 | 50 |
| 4.2 | 单列深沟球轴承参数化 | 51 |
| 4.2.1 | 单列深沟球轴承基本参数 | 51 |
| 4.2.2 | 单列深沟球轴承程序编制 | 52 |
| 4.3 | 双列深沟球轴承程序编制 | 55 |
| 4.4 | 有装球缺口深沟球轴承程序编制 | 58 |
| 4.5 | 有止动槽深沟球轴承程序编制 | 60 |
| 4.6 | 一面带防尘盖深沟球轴承程序编制 | 63 |
| 4.7 | 一面带密封圈深沟球轴承程序编制 | 66 |
| 4.8 | 对话框驱动程序 | 69 |
| 4.8.1 | 对话框驱动程序编写 | 69 |
| 4.8.2 | 对话框驱动程序运行 | 71 |
| 第五章 | 调心滚子轴承的参数化绘图 | 73 |
| 5.1 | 双列调心滚子轴承参数绘图 | 73 |
| 5.1.1 | 双列调心滚子轴承主要参数 | 73 |
| 5.1.2 | 主要定位点的确定 | 73 |
| 5.1.3 | 主要绘图过程确定 | 75 |
| 5.1.4 | 双列调心球轴承的 LISP 程序 | 77 |

| | | |
|------------|----------------------|------------|
| 5.1.5 | LISP 程序调试与结果 | 80 |
| 5.2 | 单列调心滚子轴承的程序设计 | 80 |
| 5.2.1 | 作图点设计 | 81 |
| 5.2.2 | 程序设计 | 82 |
| 5.2.3 | DCL 文件 | 84 |
| 5.3 | 对话框程序与驱动编程 | 84 |
| 5.3.1 | 对话框程序 | 84 |
| 5.3.2 | 对话框驱动程序 | 86 |
| 5.3.3 | 程序运行 | 87 |
| 第六章 | 其他类型球轴承的参数化绘图 | 88 |
| 6.1 | 调心球轴承参数化绘图 | 88 |
| 6.1.1 | 调心球轴承图形与参数 | 88 |
| 6.1.2 | 二维图形关键点确定 | 89 |
| 6.1.3 | 部分编程程序说明 | 89 |
| 6.1.4 | 圆柱孔调心球轴承的 LISP 程序 | 91 |
| 6.1.5 | LISP 程序调试与加载 | 93 |
| 6.2 | 外球面球轴承的参数编程绘图 | 95 |
| 6.2.1 | 编程说明 | 95 |
| 6.2.2 | 带偏心套外球面球轴承设计方案 | 95 |
| 6.2.3 | 带偏心套外球面球轴承图形与参数 | 95 |
| 6.2.4 | 二维图形关键点 | 96 |
| 6.2.5 | LISP 编程中的命令说明 | 99 |
| 6.2.6 | 带偏心套外球面球轴承的 LISP 程序 | 100 |
| 6.2.7 | 加载应用程序并绘制图形 | 105 |
| 6.3 | DCL 对话框设计 | 106 |
| 6.3.1 | 对话框结构设计思路 | 106 |
| 6.3.2 | 对话框文件 | 107 |
| 6.3.3 | 对话框驱动程序关键点 | 110 |
| 6.3.4 | 对话框驱动程序 | 112 |
| | 主要参考文献 | 118 |

第一章 绪 论

1.1 常用轴承参数化二维绘图背景

计算机辅助设计 (computer aided design, CAD) 以计算机为基础完成整个产品的设计过程。CAD 技术起步于 20 世纪中期, 随着计算机硬件技术的发展而迅猛发展, 其中美国 AUTODESK 公司开发的计算机辅助设计绘图软件 AutoCAD, 具有简单易学、使用方便、绘图精确和具有开放性等优点。现在 AutoCAD 已广泛应用于航空、航天、船舶、机械、服装、建筑、电子等领域, 其中在机械设计领域, 许多研究人员使用 AutoCAD 画零件的图纸。然而, 人们总感觉作图的效率不够高, 这是因为 AutoCAD 是一款通用绘图软件, 不具备换页功能。此外, AutoCAD 的开放性也为不同的使用者留出了广阔的空间, 并且提供了许多的开发工具, 其中 AutoLISP 是其中最方便的一个, 人们可以利用 AutoLISP 开发出具有专业特色的高效率的应用软件。所以, 人们希望了解 AutoLISP 的程序结构, 掌握 AutoLISP 程序设计方法, 进而开发出适合本专业特点的应用 CAD 系统。

绘图中参数化绘图越来越显示出它的重要性, 草图设计、详细设计、参数化设计起到关键的作用, 它能够准确地表达各个尺寸、元素之间的不同的约束关系。对于某些零件图的绘制, CAD 的标准功能已经无法满足绘图的需求, 在标准库中很难找到与其对应的绘制方法, 需要工程技术人员花大量的时间去绘制相关的设计图纸, 设计工作的重复性较大, 工作效率比较低。因此对此类零件需要进行参数化绘图。

本书以外圈无挡边圆柱滚子轴承、内圈无挡边圆柱滚子轴承、内圈单挡边圆柱滚子轴承和外圈单挡边圆柱滚子轴承参数化绘制标准件零件图作为研究对象, 运用 LISP 语言编写相关的图像生成程序, 利用 DCL 对话框控制语言设计用于用户输入参数的对话框界面; 用 LISP 语言编写的生成程序来调用对话框, 系统自动完成绘制图形的功能。

竞争激烈的当下, 尤其是在制造业方面, 如果机械设计人员能够准确快速地绘制出滚动轴承零件的相关图纸, 那么在设计这方面所花的时间将大大减少, 缩短产品设计的周期。应用参数化绘图可以帮助设计人员从繁重的绘图过程中解脱出来, 让设计人员在新产品的设计和构思上放更多的时间, 提高工作效率。

1.2 软件设计方法及实现步骤

应用 Visual LISP 语言编制的参数化绘图程序包括设置图层、计算点的位置和绘图。设置图层用于建立图层, 设置线型的颜色、图线的线型和图线宽度; 绘图用于绘制基本图形。通过这个程序可以让用户在对话框中选择所需要绘制轴承的类型, 然后输入必要的特征参数后, 程序自动绘制出轴承的零件图。

(1) 以相关机械设计手册为依据,以滚动轴承为研究对象,查阅国家标准 GB/T 283—1994 和 GB/T 285—1994。



(2) 掌握圆柱滚子、圆锥滚子、调心球、深沟球等轴承的分类和结构形式;分析和确定圆柱滚子轴承的特征,确定滚动轴承的特征数。

(3) 确定滚动轴承的关键参数,并用关键参数去表示其他的所有参数。

(4) 应用 AutoLISP 语言编写程序,根据已经建立好的数学模型和特征参数,计算出所有的其他参数,完成参数化绘图的程序。

(5) 应用 DCL 语言来完成用户交互界面的设计,使用户在界面上可以选择自己需要类型的滚动轴承,并可以填写特征参数的具体数值。

(6) 应用 Visual LISP 语言编写完成参数化绘图程序的驱动,实现 DCL 对话框与 Visual LISP 的连接。

(7) 联机调试。Visual Lisp 提供了监视窗口、检验窗口、中断和继续执行、程序调试按钮进行加载,加载完程序以后单击“调试”→“立即停止”,这种方式可以用在初次调试程序中,之后的程序调试可以选择在程序段后面加上断点来调试。单击按钮回到 CAD 界面在命令提示行输入程序名称回车,进行程序调试、命令跟踪、跟踪堆栈、跟踪窗口等一系列调试用窗口和工具。利用以上的窗口和工具可以迅速地查找程序错误,大大减少了程序调试所需的时间。

第二章 圆柱滚子轴承的参数化绘图

圆柱滚子轴承种类形式很多,采用 AutoLISP 完全对所有圆柱滚子轴承进行参数化绘图编程不太可能,在此以 N 型、NF 型、NU 型、NJ 型、NN3000 型、NNU4900 等为例阐述圆柱滚子轴承的参数化绘制过程。

参数化绘图就是将工程图中的图形和一组参数相联系,由这组参数表示的约束条件来确定相应零件的图形即所谓的根据参数自动生成零件图形。为此就要进行相应的程序设计,对于圆柱滚子轴承的参数化绘图程序设计,遵循步骤如下。

(1) 对所需要绘制的圆柱滚子轴承类型进行相关性分析。分析圆柱滚子轴承的结构特征,确定圆柱滚子轴承的参数。通常绘制一个零件的图形,需要很多尺寸数据,但是不可能把所有的数据都作为参数,这时就需要分析这个零件,确定既能体现零件的结构特征,又能推导出其他尺寸数据的基本数据作为绘图的特征参数,特征参数越少越好。


(2) 根据圆柱滚子轴承的绘图参数,设计用户交互界面(DCL)。对于参数化绘图程序来说,对话框主要考虑绘图参数的输入和相关控件的布局,为了增加程序的可读性,对话框中还要配有图像控件,用幻灯片来显示各绘图参数的含义。

(3) 按照图形需要,编写绘图程序。用 DCL 语言定义的对话框只是一个界面描述,不能独立运行,只有用 AutoLISP 的函数编写驱动程序来驱动,才能实现指定的功能,获取用户输入的绘图参数。根据获取的绘图参数,按照图形需要,进行相关的坐标转换,编写相应的图形绘制程序。

(4) 加载相关程序,绘制所需图形。在编辑器下编辑好的 AutoLISP 程序,必须经过加载方能使用。加载方法如下:

① 将编制的程序所有文件复制到 AutoCAD 安装目录下某文件夹。

② 打开 AutoCAD “工具”菜单,选择最下面的“选项”弹出如图 2.1 所示对话框。

③ 单击 **支持文件搜索路径** 前的  ,再单击右边的 **添加(A)...** 按钮,再单击 **浏览(B)...** 按钮,之后选择 AutoCAD 安装目录下复制好的程序所有文件所在文件夹。

④ 单击“应用”按钮,再单击“确定”按钮。

⑤ 返回 AutoCAD 主界面,单击“工具”菜单,选择“加载应用程序”,打开 AutoCAD 安装目录下的程序所在文件夹,把文件夹中的所有 Lisp 程序选中(先点选一个,再按 Ctrl+A),单击 **加载(L)** 按钮待出现加载成功后,单击 **关闭(C)** 按钮。

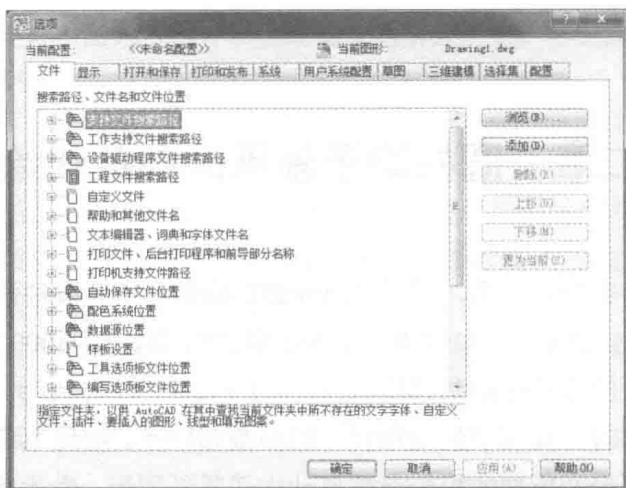


图 2.1 选项对话框

⑥ 返回 AutoCAD 主界面，命令行中输入程序名字，单击 Enter 键运行。

2.1 外圈无挡边圆柱滚子轴承参数化绘图

2.1.1 外圈无挡边圆柱滚子轴承参数

外圈无挡边圆柱滚子轴承参数如下。

- (1) 内径：轴承内圈直径，用 d 表示。
- (2) 外径：轴承外圈直径，用 d_1 表示。
- (3) 宽度：轴承宽度，用 b 表示。
- (4) 滚子最高点直径：圆柱滚子轴承最高点的直径，用 ew 表示。
- (5) 内挡圈直径：圆柱滚子轴承内挡圈直径，用 d_2 表示。
- (6) 内圈圆角：圆柱滚子轴承内圈的圆角，用 r 表示。
- (7) 外圈圆角：圆柱滚子轴承外圈的圆角，用 r_1 表示。

用 LISP 程序绘制外圈无挡边圆柱滚子轴承需查阅国家标准圆柱滚子轴承 GB/T 283—1994，其部分主要参数如表 2.1 所示。

表 2.1 外圈无挡边圆柱滚子轴承参数

| 编号 | d | d_1 | b | ew | d_2 | r | r_1 |
|----|-----|-------|-----|------|-------|-----|-------|
| 0 | 15 | 35 | 11 | 19.3 | 26.4 | 0.6 | 0.3 |
| 1 | 25 | 52 | 15 | 46.5 | 34.7 | 1 | 0.6 |

外圈无挡边圆柱滚子轴承国家规定画法如图 2.2 所示。

2.1.2 程序流程

圆柱滚子轴承的绘图流程图如图 2.3 所示。

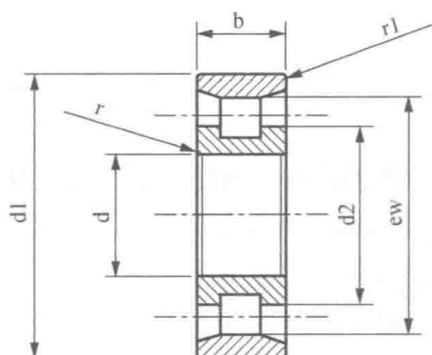


图 2.2 外圈无挡边圆柱滚子轴承标准画法及参数

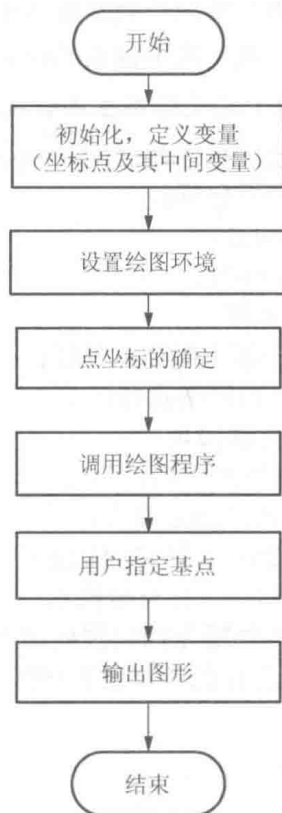


图 2.3 绘图基本流程

(1) 开始。定义绘图函数名称。

(2) 变量定义。变量定义主要是定义坐标点及程序中的中间变量。

(3) 绘图环境设置。为避免绘图时出现错误，绘图前关闭对象捕捉、回显等基本操作。

(4) 坐标点的确定。坐标点确定包括以下两种类型。

① 输入点。主要定位基点，需用户在 AutoCAD 界面指定的图形基点。

② 计算坐标点。根据用户输入或确定的定位基点和特征变量确定其他点。

(5) 调用绘图程序。根据上面点坐标的计算，调用绘图命令进行圆柱滚子轴承图形的绘制，调用的绘图命令主要有直线 (line)、圆角 (filled)、旋转 (rotate)、镜像 (mirror)、填充剖面线 (bhatch) 及中心线等。

(6) 尺寸标注。标注轴承的内径、外径和宽度主要参数。

2.1.3 程序说明

程序各部分的起始部分均对用户的当前作图环境设置进行保存，在程序的结束时又进行了恢复作图设置的操作，这使得程序中图层设置不会与用户的图层设置产生矛盾或干涉，从而保证用户在使用前后的连贯性。

由于所绘制的标准件是添加到其他主体图形上的，本程序的所有分程序的第一个参数是基准点。图形基准点选择的是否合理，决定了绘图效率的高低，一个合理的基准点

往往可以使图形的绘制准确和快速。在对圆柱滚子轴承编程时，基准点 p 选择了轴承的右侧中点。圆柱滚子轴承的绘制还需要另外的三个参数，即内径、外径和宽度。

2.1.3.1 定义绘图主函数及其变量

(1) 绘图主函数名称：yzgz。

(2) 变量包括：

d——内径；

d1——外径；

b——宽度；

ew——滚子最高点外径；

d2——内挡圈直径；

r——内圈圆角；

r1——外圈圆角；

p——用户指定基点；

p1~p20——计算坐标点；

c1~c11——计算坐标点。

因为该外圈无挡边圆柱滚子轴承，具有上下对称的特点，因此只定义局部点 p1~p20，其点坐标的具体位置如图 2.4 所示。

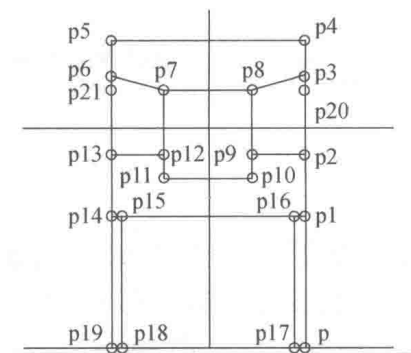


图 2.4 N 型圆柱滚子轴承点坐标位置图

图中：p——设置为图形的基点，其他的点都通过此点来确定他们的位置；

p1——圆柱滚子轴承内径与右边的交点，通过 p 点来确定位置；

p2——通过 P 点确定，水平可以确定内挡圈这条线上的点；

p3——通过 p8 点确定，p8p3 与 X 轴夹角为 15°，知道 p8p20 的距离，可用三角函数求得 p3 点的位置；

p4——通过 p 点确定，距离 p 点为轴承外径的一半；

p20——轴承滚子最高点，通过 p 点确定，距离为 ew 的一半。

2.1.3.2 各点坐标值的确定

设 $a = (d1 - d) / 2$ ，见表 2.2。

表 2.2 各点坐标确定

| 计算点 | 参考点 | 距离 | 角度 | 计算点 | 参考点 | 距离 | 角度 |
|-----|-----|--|---------|-----|-----|------------|-------|
| p1 | p | 0.5d | 0.5pi | p12 | p13 | 0.5b-0.25a | 0 |
| p2 | p | 0.5d2 | 0.5pi | p13 | p2 | b | pi |
| p3 | p8 | $(0.5b-0.25a) / \cos 75^\circ / \cos 15^\circ$ | pi/12 | p14 | p1 | b | pi |
| p4 | p | 0.5d1 | 0.5pi | p15 | p14 | r | 0 |
| p5 | p4 | b | pi | p16 | p1 | r | pi |
| p6 | p7 | 0 | 11pi/12 | p17 | p | r | pi |
| p7 | p21 | 0.5b-0.25a | 0 | p18 | p19 | r | 0 |
| p8 | p20 | 0.5b-0.25a | pi | p19 | p | b | pi |
| p9 | p2 | 0.5b-0.25a | pi | p20 | p | 0.5ew | 0.5pi |
| p10 | p8 | 0.5a | 1.5pi | p21 | p20 | b | pi |
| p11 | p7 | 0.5a | 1.5pi | | | | |

2.1.3.3 根据程序流程图编制程序

(1) 定义绘制圆柱滚子轴承的命令。

(defun c:yzgz (/ e0 e1 xzj d0...)) ; 定义绘制圆柱滚子轴承命令的命令名为 yzgz。

(2) 保存当前图层设置并设置捕捉方式。由于每一个程序都要使用，可以将这些操作定义成一个函数，这样调用起来就比较方便。

```
((defun f_gettc () ; 获得图层信息函数并赋新值
  (setq os (getvar "osmode")) ; 获取当前目标捕捉方式并赋给变量 os
  (setq sc (getvar "cmdecho"))
  (setq pre (getvar "luprec"))
  (setq clay (getvar "clayer")) ; 获取当前图层信息，并赋给变量 clay
  (setvar "osmode" 0) ; 关闭对象捕捉
  (setvar "cmdecho" 0)
  (setvar "luprec" 1)
  )
```

(3) 接受用户输入参数，并赋值给相应的变量。

```
(setq p (getpoint "输入基点: "))
(if (= p nil)(setq p (list 0 0)))
(setq d (getdist "输入内径: "))
(setq d1 (getdist "输入外径(D): "))
(setq b (getreal "输入宽度: "))
```

(4) 确定绘图的其他参数。由于圆柱滚子轴承是标准件，可以通过 cond 函数来确定其他的参数 ew、d2、r 和 r1。程序如下：

```
(cond (测试表达式 1 结果表达式 1) [(测试表达式 2 结果表达式 2)] ...)
```

该函数从第一个子表起，计算每一个子表的测试表达式，直至有一个子表的测试表达式成立为止，然后计算该子表的结果表达式，并返回这个结果表达式的值。

例如, 当 i 小于等于 1 时, $n=1$; 小于等于 2 时, $n=4$; 小于等于 3 时, $n=10$; 其他情况下 $n=100$ 。用 `cond` 函数实现变量 n 和 i 之间以上关系的源代码如下:

```
(setq n(cond ((<= i 1) 1)
             ((<= i 2) 4)
             ((<= i 3) 10)
             (t 100) ) )
```

`Cond` 函数在程序中的应用, 代码如下

```
(setq ew (cond ((and (= d 15) (= d1 35)(= b 11)) 29.3)
              ( (and (= d 17) (= d1 40)(= b 12))33.9)
              ((and (= d 20) (= d1 42)(= b 12))36.5)
              ((and (= d 20) (= d1 47)(= b 14))41.5)
              ((and (= d 20) (= d1 47)(= b 18))41.5)
              ((and (= d 20) (= d1 52)(= b 15))45.5)
              ((and (= d 20) (= d1 52)(= b 21))45.5)
              ((and (= d 25) (= d1 52)(= b 15))46.5)
              ((and (= d 25) (= d1 52)(= b 18))46.5) ) )
(setq d2 (cond((and (= d 15) (= d1 35)(= b 11))22)
             ((and (= d 17) (= d1 40)(= b 12))25.5)
             ((and (= d 20) (= d1 42)(= b 12))28.3)
             ((and (= d 20) (= d1 47)(= b 14))29.7)
             ((and (= d 20) (= d1 47)(= b 18))29.7)
             ((and (= d 20) (= d1 52)(= b 15))31.2)
             ((and (= d 20) (= d1 52)(= b 21))31.2)
             ((and (= d 25) (= d1 52)(= b 15))34.7)
             ((and (= d 25) (= d1 52)(= b 18))34.7)))
(setq r (cond((and (= d 15) (= d1 35)(= b 11))0.6)
            ((and (= d 17) (= d1 40)(= b 12))0.6)
            ((and (= d 20) (= d1 42)(= b 12))0.6)
            ((and (= d 20) (= d1 47)(= b 14))1)
            ((and (= d 20) (= d1 47)(= b 18))1)
            ((and (= d 20) (= d1 52)(= b 15))1.1)
            ((and (= d 20) (= d1 52)(= b 21))1.1)
            ((and (= d 25) (= d1 52)(= b 15))1)
            ((and (= d 25) (= d1 52)(= b 18))1) ) )
(setq r1 (cond((and (= d 15) (= d1 35)(= b 11))0.3)
             ((and (= d 17) (= d1 40)(= b 12))0.3)
             ((and (= d 20) (= d1 42)(= b 12))0.3)
             ((and (= d 20) (= d1 47)(= b 14))0.6)
             ((and (= d 20) (= d1 47)(= b 18))0.6)
             ((and (= d 20) (= d1 52)(= b 15))0.6)
             ((and (= d 20) (= d1 52)(= b 21))0.6)
             ((and (= d 25) (= d1 52)(= b 15))0.6)
             ((and (= d 25) (= d1 52)(= b 18))0.6) ) )
```

(5) 确定关键点坐标, 程序用 `polar` 函数来完成点坐标的确定。

```
(setq p1 (polar p (* 0.5 pi) (* 0.5 d))) ;关键点的确定
(setq p16 (polar p1 pi R))
(setq p14 (polar p1 pi B))
```

```
(setq c2 (polar p pi (* 0.5 b)))
(setq c3 (polar c2 (* 0.5 pi) (+(* 0.5 d1)5)))
```

(6) 绘制图形。

① 图层参数确定。

```
(command "layer" "m" "0" "c" 7 "0" "l" "continuous" "0" "lw" 0.36 "0" "" )
```

其中, "layer"为命令; "m"表示新建图层; "0"表示图层名为 0; "c"表示修改颜色; 7 表示颜色为 7 (白色); "0"表示将线型 continuous 应用到图层 0; "lw"表示设置线条宽度为 0.36; "0"表示将线条宽度为 0.36 应用到图层 0; ""表示图层将建立完成。

② 图形绘制。

(command "line" p p1 p2 "") ; 调用命令"line", 并根据参数 p、p1、p2 绘制线段, 在程序中的应用, 代码如下:

```
(command "line" p p1 p4 "")
(command "line" p5 p14 p19 "")
(command "line" p5 p4 "")
(command "line" p3 p8 "")
```

③ 图形剖面线绘制。选取截面内部的点然后用命令 (command "bhatch" "p" "ansi31" "0.25" "0" 内部点 "") 绘制剖面线。

p: 拾取点的方式。 ansi31: 剖面线的标准名称。

0.25: 剖面线的比例因子。 0: 剖面线的角度。

在程序中的应用, 代码如下:

```
(setq pm1 (polar p3 pi (* b 0.5)))
(setq pm2 (polar p16 (* pi 0.5)2))
(command "zoom" "c" p14 150)
(command "bhatch" "p" "ansi31" "0.2" "0" pm1 "")
(command "bhatch" "p" "ansi31" "0.2" "90" pm2 "")
```

④ 尺寸标注。圆柱滚子轴承尺寸标注有内径标注、外径标注和宽度标注。圆柱滚子轴承内径的标注要用到长度标注命令, 需要选择三个点, 用命令 (command "dimaligned" 点1 点2 "h" 点3), 点1、点2 是第1、2 条尺寸线的起点的位置, "h" 尺寸线的方向, 可以是水平方向、也可以是垂直方向 (v) 还可以指定为其他方向 (rotated)。点3 为尺寸线的位置确定点, 可以是一个变量, 也可以是一个点的坐标。变量描述了点3 与点2 的距离, 坐标定的点在标注线上。外径和宽度的标注和内径是一样。

⑤ 绘图倒圆角。

```
(command "fillet" "r" r1 "fillet" 点1 点2)
```

其中 "fillet" 为 Lisp 中倒角的命令; "r" 为以圆角半径的方式来倒圆角; "r1" 为圆角半径, 其必须在前面有定义否则无法倒圆角; 点1、点2 分别是两条相交线上的点, 注意 (点1 点2) 这两个点是由两条直线交点为基础点来定义的。

在编写圆柱滚子轴承是用以下的命令来完成倒圆角命令的:

```
(command "fillet" "r" r1 "fillet" c1 c2)
(command "fillet" "r" r1 "fillet" c3 c4)
(command "fillet" "r" r1 "fillet" c5 c6)
```



```
(command "fillet" "r" r1 "fillet" c7 c8)
```

- ⑥ 图形元素进行操作，使其显示在用户所需的位置。构成选择集。

```
(setq e0 (entlast))
```

```
:
```

```
(if (= e0 nil)(setq e1 (entnext))(setq e1 (entnext e0)))
```

```
(setq xzj (ssadd))
```

```
(while (/= e1 nil)
```

```
(ssadd e1 xzj)
```

```
(setq e1 (entnext e1)) )
```

- (7) 对选择集操作。

- ① 旋转命令"rotate":

```
(command "rotate" xzj "" p ang)
```

该命令对选择集 xzj 进行旋转，以 p 为基点，旋转 ang。

- ② 移动命令"move":

```
(command "move" xzj "" p pt)
```

该命令对选择集 xzj 进行移动，以 p 为基点移动到 pt 点。

- ③ 为避免连线时因对象小而发生错误，因此连线前将绘图区域放大，放大命令为"zoom"其程序如下：

```
(command "zoom" "w"点1 点2 点3 "") 点1 点2 点3 确定放大区域
```

```
(command "zoom" "w" c3 p1 p14 "")
```

- ④ 前面程序做出来的图为圆柱滚子轴承的一半，可应用镜像命令“mirror”来坐出一个完整的图，其程序如下：

```
(setq s(ssget "X"))(command "mirror" s ""点1 点2 "")
```

其中 (setq s (ssget "X")) 这段程序代码的意思是选取前面的所有的图元建立选择集 s，注意"X"是选择集操作函数的值。点1 点2 所确定的直线是我们镜像的轴线。在程序中的程序段如下：

```
(setq s1(ssget "x"))
```

```
(command "mirror" s1 "" p p19 "")
```

- (8) 恢复保存的图层信息。

```
(defun f_backtc () ;还原图层信息函数
```

```
(setvar "osmode" os)
```

```
(setvar "cmdecho" sc)
```

```
(setvar "luprec" pre)
```

```
(setvar "clayer" clay))
```

2.1.3.4 程序完整代码

```
(defun f_gettc () ;获得图层信息函数并赋新值
```

```
(setq os (getvar "osmode"))
```

```
(setq sc (getvar "cmdecho"))
```

```
(setq pre (getvar "luprec"))
```

```
(setq clay (getvar "clayer"))
```

```
(setvar "osmode" 0)
```

```
(setvar "cmdecho" 0)
```

```
(setvar "luprec" 1))
```