

# 计算机辅助设计控制系统的新工具 ——CADCS 智能软件包

梁义芝 李国栋

(大连舰艇学院软件中心) (大连海运学院计控系)

刘云飞

(大连舰艇学院电教中心)

**摘要** 本文将人工智能理论用于控制系统的计算机辅助设计,建立了控制系统分析与设计专家系统,然后在此专家系统的基础上,进一步实现了控制系统分析与设计应用程序的自动生成,将常规 CADCS 软件包发展为 CADCS 智能软件包,为用户进行控制系统的分析与设计提供了更为有效、方便和可靠的新工具,文中介绍了 CADCS 智能软件包的组成及其各部分的实现原理,最后通过实例给出了智能软件包的工作过程。

**关键词:** 控制系统计算机辅助设计, 专家系统, 程序自动生成, 智能软件包

## 1 引言

从 60 年代开始,计算机逐渐引入到包括控制系统设计的工程设计领域,形成了 CADCS(计算机辅助设计控制系统)这样一门综合性很强的先进技术,其发展大致经过了三个阶段<sup>(7)</sup>:

- (1) 针对具体问题,编制相应的程序代替手工计算和作图。
- (2) 出现了解决一类问题的问答式通用程序,形成 CADCS 程序库。
- (3) 在 CADCS 程序库的基础上,又发展了 CADCS 程序包,程序包中增加了管理程序。

CADCS 程序包在一定程度上方便了用户使用,其管理程序利用菜单将所有功能管理起来,用户可根据需要进行选择,但是对于不经常进行控制系统设计或没有将控制理论应用于具体设计经验的用户来说,使用这种软件包难于较好地设计控制系统。随着 CADCS 软件包功能的不断增加,软件包日趋庞大,要保持设计步骤好于现存软件包所提供的环境,所需要的知识、理解力和对每一细节的注意力都使用户望而生畏,也许超过了他们所具备能力的极限。因此,用人工智能的方法管理、使用和完善 CADCS 已经开始引起人们的重视。本文提出将常规 CADCS 软件包发展为 CADCS 智能软件包,为人们提供了一种计算机辅助设计控制系统的新工具。



图 2.1 常规 CADCS 软件包构成简图

## 2 CADCS 智能软件包的结构

常规 CADCS 软件包一般有如图 2.1 的结构。

其中基础程序库中包含有完成控制系统分析与设计所需各算法子程序,应用程序即是某一具体分析与设计方法的实现程序(通过对基础程序的调用完成),而用户通过管理程序来选择不同的应用程序,以完成系统分析与设计。

CADCS 智能软件包结构见图 2.2,在 IBM-PC 机上用 Gclisp 语言编写,而控制系统分析与设计的具体算法是由 FORTRAN 语言来实现的。

图 2.2 中,智能软件取代了常规软件包中的管理程序与应用程序层,若用户进行控制系统设计,不必自己选择设计方案,只需将待设计系统的有关信息送给智能软件,即可得到由智能软件即时生成的完成此设计的应用程序,执行此应用程序便可设计出满意的控制系统并可进行系统仿真,其示意图如图 2.3。

①收到本文的时间是1988年11月8日。

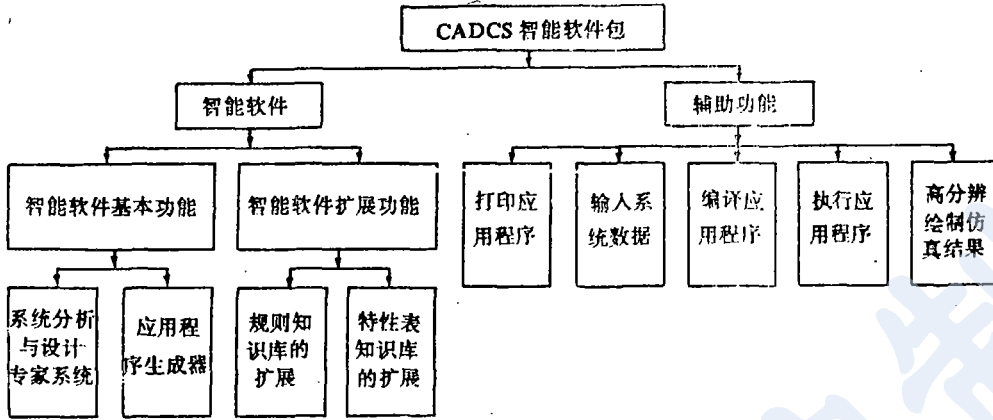


图 2.2 智能软件包组成

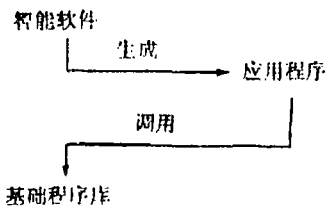


图 2.3 智能软件包工作示意图

CADCS 智能软件包便于系统功能的增加和改进,若增加控制系统分析与设计新方法,只需将算法实现程序装入基础程序库,并由智能软件扩展功能扩展知识库即可,而不必象常规软件包那样除增加基础程序库外,还须修改管理程序和编写应用程序等。

### 3 智能软件基本功能介绍

本软件包中的控制系统分析与设计专家系统和应用程序生成器组成了智能软件基本功能部分。它要求用户输入待分析或设计控制系统的有关信息,如噪声、反馈状态个数等,根据这些信息由系统分析与设计专家系统选定分析与设计方案,并将之具体化为完成此方案的算法实现步骤,再由应用程序生成器将算法实现步骤程序化为相应的应用程序。从应用程序生成的角度来讲,专家系统完成应用程序的整体组织(它规定了各基础子程序的调用顺序),程序生成器完成应用程序的局部组织(它规定了各基础子程序的调用方法)。两者的关系如图 3.1 所示。

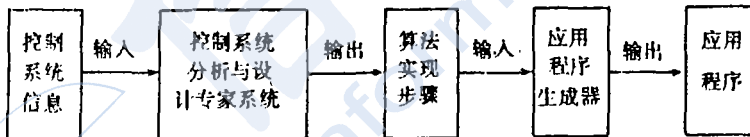


图 3.1 应用程序生成过程示意图

#### 3.1 控制系统分析与设计专家系统

常规 CADCS 软件包提供了多种控制系统的分析与设计方法,如何针对本系统的情况选择适当的设计方法及方法选定后如何利用 CADCS 软件包完成控制系统的分析与设计任务,这对缺乏经验和知识的非专家用户来讲是比较困难的。为此我们编制了控制系统分析与设计专家系统,其组成见图 3.2。

本专家系统对具有不同特性的控制系统,可利用其专家知识和经验帮助用户选择分析与设计方案,以及用此方案完成分析与设计任务的具体实现步骤。

##### 3.1.1 知识库组织

控制系统分析与设计专家系统的知识库分为控制系统分析与设计方法选择知识库和控制系统分析与设计算法选择知识库。将知识库分级是加快推理进程的一种切实可行的方法,且可使知识库层次清晰,易于管理,特别适合在微机上实现。

方法选择知识库是由这样形式的规则组成的: if a then b, a 为前件, b 为后件。后件是具体的设计方案名字,前件是选择此方案应满足的条件。例如:有这样的五条规则:

- (1) (如果(部分状态反馈)(没有噪声))(则(次优控制规律))

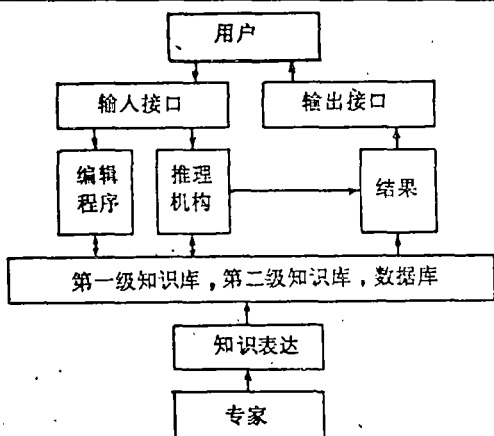


图 3.2 专家系统组成示意图

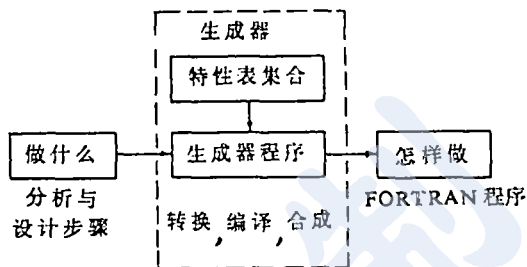


图 3.3 应用程序生成器原理图

- (2) (如果(全状态反馈)(存在白噪声)(所有状态可测量))(则(线性二次型高斯控制规律))
- (3) (如果(全状态反馈)(无噪声)(部分状态可测量))(则(极点配置法 1))
- (4) (如果(全状态反馈)(无噪声)(所有状态可测量)(给定权矩阵))(则(线性二次型最优控制))
- (5) (如果(全状态反馈)(无噪声)(所有状态可测量)(给定极点数据))(则(极点配置法 2))

算法选择知识库亦是规则知识库，由于系统数据输入方式的不同，各设计方案对应有不同的算法实现过程，这些知识存于算法选择知识库。

### 3.1.2 推理机

系统分析与设计专家系统提供正、反两种推理机制，都是确定性推理，采用顺序匹配方式。利用知识库的知识及用户提供的事实，推理机可给出系统分析与设计的具体步骤，用户可按此步骤进行控制系统分析与设计。

## 3.2 软件包应用程序生成器

用户可按专家系统给出的步骤，利用常规软件包完成系统设计，但需反复进行菜单选择，送入系统数据、阶次等，具体实现起来也很麻烦。为此我们希望控制系统分析与设计专家系统所做的工作只是第一步，作为软件包应用程序生成器的基础，应用程序生成器则将此步骤程序化，形成一个完整的 FORTRAN 程序，用户运行此程序即可完成上述分析设计任务。

软件包应用程序生成器原理如图 3.3。图中特性表集合即为生成器的知识库，生成器利用其中的知识将专家系统给出的分析与设计步骤经过转换、编译、合成，形成 FORTRAN 程序。

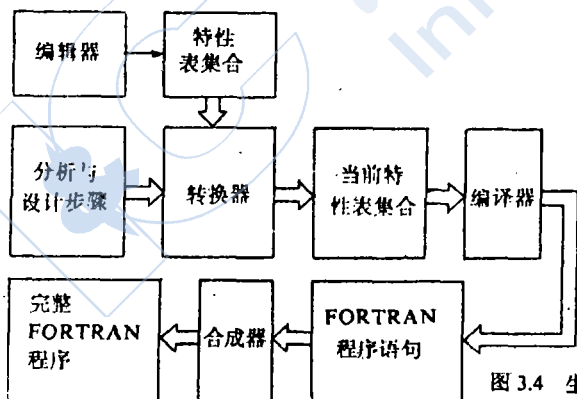


图 3.4 生成器系统结构图

应用程序生成器的系统结构见图 3.4。由编辑器、特性表集合、转换器、编译器与合成器五部分组成。转换器将各分析与设计步骤转换为与之对应的特性表，编译器则将各特性表编译成为独立的 FORTRAN 语句，再由合成器将之合成为一个完整的 FORTRAN 程序。

(1) 编辑器：使用编辑器进行特性表的编辑工作。Gclisp 所提供的 GMACS 编辑状态，DOS 下的编辑状态，及本文所提供的特性表维护手段(即智能软件扩展功能)，都可做为编辑器使用。

(2) 特性表集合：如前所述，每一分析与设计算法都对应一个基础子程序，应用程序则通过调用

基础子程序来完成系统分析与设计任务, 因此应用程序基本上是由对基础子程序的调用语句及输入输出语句组成。就是说应用程序包含几个相同的基本部分, 结构清晰。通过对各基础子程序的分析, 我们将其调用方法及辅助内容归纳为七部分, 做为七个特性组成一张特性表如下:

特征	特征值
funame	基础子程序名
sfufile	所调用子程序文件名集合
comarean	基础子程序所用公用区名集合
filename	基础子程序所用数据文件名集合
fread	读数据部分
calculate	计算部分(调用部分)
stop	结束部分

这七个特性完整地说明了基础子程序的使用方法, 易于自动编程, 易于理解, 亦易于增加和修改知识。

实际上控制系统分析与设计专家系统给出的每一分析设计步骤都是由一个基础子程序完成的, 即每一步骤对应一个特性表。软件包中所有基础子程序的特性表即成为特性表集合, 应用程序即是应用程序生成器作用于特性表集合所得。

(3) 转换器: 转换器实际上有两项任务, 一是从控制系统分析与设计专家系统的结果数据库中提取分析与设计步骤; 二是检查这些步骤是否有基础子程序特性表与之对应。若都可对应则将所对应的特性表取出, 形成当前特性表集合, 否则给出错误信息。

(4) 编译器: 编译器是程序生成器中较复杂的部分, 针对特性表中六部分(除去基础子程序名)内容的不同, 有三种编译器与之对应: 对应 sfufile 的一种编译器较简单, 只将其值译成应用程序的解释部分, 供用户编译 FORTRAN 程序时用; 对应 comarean 和 filename 的编译器, 首先要将当前特性表集合中各特性表的 comarean 特征值和 filename 特征值分别组合起来, 去除冗余, 形成应用程序的公用区名集合和所用数据文件名集合, 然后检查公用区名和数据文件名的合理性, 最后根据公用区名集合形成应用程序的公用区语句, 根据数据文件名集合形成对数据文件名的读语句及说明语句; 对应 fread, calculate 和 stop 三部分的编译器则形成应用程序的读数据、基础子程序调用及结果输出部分。

(5) 合成器: 将编译器编译出的内容合成为一个完整的 FORTRAN 程序。

需要说明的一点, 程序生成器生成的应用程序没有对整型和实型变量的类型说明, 程序使用时用其隐含方式。

所生成的应用程序为 FORTRAN 源程序, 存入用户指定文件名的文件中。用户可回到智能软件包辅助功能来编译, 执行所得应用程序, 完成系统设计。

#### 4 应用实例

图 4.1 为卫星单轴姿态控制系统示意图<sup>(8)</sup>。该系统简化后运动方程可写为  $J\ddot{\theta} = M_c$ , 其中  $M_c$  是由喷嘴 A 和 B 所产生的外加力矩,  $J$  是转动惯量,  $\theta$  是卫星的一个姿态角, 它由喷嘴 A 或 B 控制其偏转。写出其相应状态方程为

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

其中

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix}, \quad u = \frac{M_c}{J}, \quad y = \theta$$

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad C = [1 \quad 0]$$

若要求等效的闭环连续系统具有一对共轭极点为

$$s_{1,2} = -1.8 \pm j 3.1177$$

试用极点配置法设计此控制系统。

若系统初值为  $x_0 = \begin{bmatrix} 5 \\ 0 \end{bmatrix}$ , 初始输入为  $u_0 = 0$ , 试仿真设计结果。

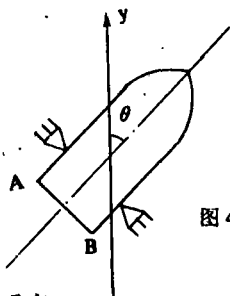


图 4.1 卫星单轴姿态控制系统示意图

② 无噪声。

③ 部分状态可测量。

用户回答成立, 则假设为真。

专家系统提问用户数据输入方式, 回答为输入连续状态方程, 则给出如下系统设计步骤:

① 判断连续系统的可控可观性。

② 连续状态方程  $\rightarrow$  离散状态方程。

③ 实现极点配置。

④ 求观测器值。

⑤ 判断所设计闭环系统的稳定性。

⑥ 仿真闭环系统。

(2) 程序生成器将上述步骤转换为 FORTRAN 应用程序

系统提请用户输入存放 FORTRAN 程序的文件名, 将所生成程序存入其中, 结果见附录。

运行此程序所得控制器值为

$$L = [0.107669E + 02 \quad 0.356158E + 01]$$

观测器值为

$$K = [0.200000E + 01 \quad 0.100000E + 02]$$

仿真结果略。

解 使用本智能软件包完成系统设计的步骤如下:

(1) 进入系统分析与设计专家系统

用户选用反向推理机并提出假设为: 使用极点配置法 1 设计控制系统。

专家系统向用户提问以下条件是否成立:

① 全状态反馈。

## 参 考 文 献

- 1 Goldhill Computer公司. Golden Common Lisp Reference Manual. 1984
- 2 Gerarler W E. Expert System: Limited But Powerful, IEEE Spectrum, August, 1983
- 3 James H Taylor. An Expert System Scenario for Computer-Aided Control Engineering
- 4 赵瑞清. 专家系统原理. 气象出版社, 1987
- 5 何华灿. 人工智能导论. 航空工业教材编审组, 1983
- 6 孙楚辉, 张尉. 基于知识的应用软件自动生成一方法与工具. 1987年全国青年计算机工作者学术交流会论文, 1987.7
- 7 韩民, 袁曾任. 计算机辅助设计控制系统程序包管理程序. 清华大学计算机系, 1985.5
- 8 孙增圻. 计算机辅助设计控制系统. 清华大学, 1985

## 附 录

storage:2

```
c (POLE2 F15 S11 S10 S9 F3 S2 F2 S3 S1) (B1 S7 S8 F16 F4 B2 B3 B4 S4 F14)
common / d13 / h1(50), wr(50), wi(50)
common / d1 / a(50), b(50), c(50)
common / d7 / r1(50), r2(50)
common / d2 / f(50), g(50), h(50)
common / d10 / c1(50), s(50)
```

```

common / d11 / ck(50),p(50)
common d9 / x0(50),r(50),aa(61,9)
common / d8 / n,m,l,n1,n2,l,c,d
character * 2 F12
character * 2 F13
character * 2 F2
character * 2 F3
character * 2 F4
character * 2 F14
character * 2 F5
write( *,10)
10  format(1x,'n,m,l(i3)=\')
    read( *,15)n,m,l
15  format(3i3)
    write( *,75)
75  format(1x,'cigvalue file name=\')
    read( *,200) f12
    write( *,80)
80  format(1x,'pole assignment feedback data file name=\')
    read( *,200) f13
    write( *,25)
25  format(1x,'continuouse system data file name=\')
    read( *,200) f2
    write( *,30)
30  format(1x,'discreat system data file name=\')
    read( *,200) f3
    write( *,35)
35  format(1x,'pole data file name=\')
    read( *,200) f4
    write( *,85)
85  format(1x,'pole assignment estimator datd file name=\')
    read( *,200) f14
    write( *,40)
40  format(1x,'initial data file name=\')
    read( *,200) f5
200 format(a2)
    call
    fi(f1, d, n, n, f1, d, n, n, f1, d, n, n, f1, d, n, n, f1, d, n, n,
    f1, d, n, n, 'a', a, n, n, 'b', b, n, m, 'c', c, i, n, f2, 3)
    n1 = 1
    n2 = 1
    call
    coob
    n2 = 2
    call coob
    write( *,250)
250 format(1x,'sample time(e13.6)=\')
    read( *,255) t
255 format(e13.6)
    call cd1
    call
    fo(f1, d, n, n, f1, d, n, n, f1, d, n, n, f1, d, n, n, f1, d, n, n, 'f', f, n, n, 'g', g, n, m, 'h', h, l, n,
    f3, 3)
    call
    fi(f1, d, n, n, f1, d, -n, n, f1, d, n, n, f1, d, n, n, f1, d, n, n, f1, d, n, n, f1, d, n, n, f1, d, n, n,
    'r1', r1, l, n, 'r2', r2, l, n, f4, 2)
    call
    ccl if(d.eq.0) goto 400

```

(下转第29页)

