

文章编号: 1009-8119 (2003)11-0034-05

ORION 集成开发环境使用详解

李建军

(欧比特<珠海>软件工程有限公司 519080)

摘 要 介绍了 Orion 3.0 集成开发环境的基本概念和基本操作, 详细描述了从建立一个应用程序到调试, 直至完成 ROM 映像文件的全过程, 指导读者使用 Orion 3.0 IDE 开发基于 Orbita EOS 的应用程序。

关键词 S698 集成开发环境, 嵌入式操作系统, 模拟器, 硬件调试单元

Application Development in ORION - IDE

Li Jianjun

(Orbita Software Engineering Inc. Orbita TechPark, Zhuhai, Guangdong 519080)

Abstract In this article, we introduce the basic concepts and operations of the ORION 3.0 IDE environment. The procedure by which we build, debug, and create the ROM image file has been described in great detail. We expect that this shall guide the readers to easily develop the application programs based on Orbita EOS in the ORION 3.0 IDE.

Keyword S698 IDE, EOS, Emulator, DSU

1 Orion 用途

Orion 3.0 是 Linux 环境下的嵌入式集成开发环境。提供针对 Orbita EOS 的嵌入式软件开发过程中的工程项目管理、源程序编辑、编译、连接、运行、调试等功能, 同时提供各种工具协助软件的开发。

Orion 3.0 集成开发环境界面友好, 开发环境的功能及用户界面的风格同 Visual Studio 相似。利用此开发环境, 用户不仅可以调试高层应用程序, 还可以查看开发板内所有存储器资源及处理器的内部寄存器, 更加方便嵌入式应用软件开发。

2 功能

Orion 3.0 使用类似 Microsoft Visual Studio 的界面设计, 同时具备以下几项特性:

- (1) 支持 ERC32, Sailing S698 CPU。
- (2) 支持 PC386. (beta)。
- (3) 支持基于 RTEMS 4.5.0 的 Orbita EOS 实时操作系统核心。
- (4) 支持 Orbita EOS 所提供的 3 种 API: Classic, POSIX, iTRON。
- (5) 集成的操作系统配置界面。
- (6) 提供工程生成向导。
- (7) 提供工程管理, 包括向工程新建, 添加, 删除文件/类。
- (8) 编辑器支持语法高亮, 在线帮助, 上下文提示等人性化的功能。
- (9) 支持多种调试方式, 包括本地仿真调试, 本地目标

系统在线调试, 远程网络目标系统在线调试。

(10) 提供 ERC32/Sailing debug monitor。

3 运行环境

3.1 硬件设备

运行本软件所要求的硬件设备的配置:

- a) 386 或以上 CPU 的个人电脑;
- b) 128 Mbytes 或以上的 RAM;
- c) 至少 350 Mbytes 的可用硬盘空间;
- d) 具有两个串口(可选);
- e) 网卡(可选)。

3.2 支持软件

运行本软件所需要的支持软件:

- a) 操作系统: RedHat Linux 7.3;
- b) 需要安装 RedHat Linux 7.3 的开发环境套件(包含 gcc/gdb 等)。

4 使用过程

一个嵌入式应用程序的开发一般遵循如下流程图所示的步骤进行(见 35 页框图):

4.1 安装与初始化

4.1.1 Orion 安装

启动 RedHat Linux 7.3, 进入 KDE, 打开一个 Konsole:

(1) 转到 CDROM 的 mount 目录

```
[root]#cd /mnt/cdrom
```

(2) 运行 INSTALL 安装脚本

```
[root]#. /INSTALL
```

(3) 按照屏幕提示操作, 安装过程可能需要 1~3 分钟

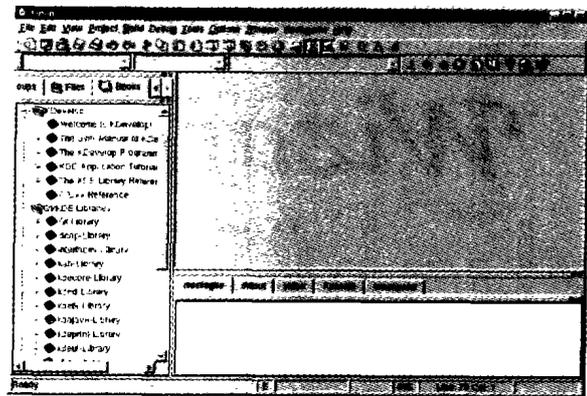
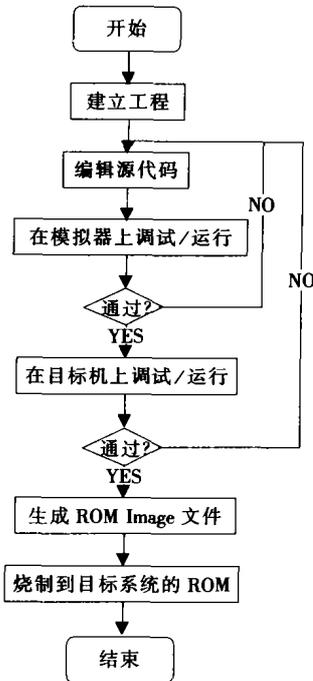


图 1 主界面

些常用的功能被做成按钮放在工具栏上, 如果不清楚具体按钮的用途, 把鼠标停留在按钮上一段时间, 将会有提示按钮的用途。

b) 视图窗口: 位于左边, 可在类视图, 文件视图, 组视图, 文档视图, 变量观察视图(调试时有效)等视图间切换。

c) 编辑窗口: 位于右边上半部, 在此编辑, 修改源代码。

d) 信息窗口: 位于右边下半部, 在此显示编译/连接信息, 输出/错误信息, 断点信息, 调用堆栈信息(调试时有效), 反汇编信息(调试时有效)等。

e) 状态条: 显示一些提示信息。

4.2.2 菜单

展开菜单, 把鼠标停留在菜单上, 状态栏会显示出菜单的功能描述。具体菜单命令和 windows 下的其他 IDE 类似。

4.2.3 工具栏

把鼠标停留在工具栏按钮上, 动态提示框会显示出工具栏按钮的功能描述, 如图 2。

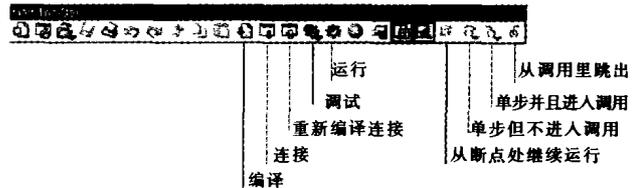


图 2 工具栏

4.3 运行

以下通过一个实例来说明使用 Orion 进行软件开发的过程。

4.3.1 工程向导

说明: 工程向导的功能是引导用户一步一步的生成一个工程, 工程是软件项目的组织单位, 一个工程包含一个或者多个源代码文件, 一个 Makefile, 以及工程设置。

点击菜单: Project>New..., 出现工程向导如图 3 所示。

选择一种类型, 这里选择 Orion C++, 然后点 Next, 再照向导的指引一步步走下去, 可以很容易地完成工程地创建。在向导引导过程中, 如果有不清楚地选项, 请保留缺省值。

Installing Orion v3.0 ...

####

#####

#####

Installation Done. Please re-login.

Type "orion" or click "Development/Orion 3.0" to run.

(4) 重新登录

[root]#exit

(5) 重新登录后, 在 KDE 的主菜单里, 选 Development/Orion 3.0, 启动 Orion 3.0

4.1.2 Debug monitor 安装

所谓 Debug Monitor, 是指被特殊设计来作为除错工具的一部分。它一般被放在 ROM 中, 通过串口或网络与除错器进行通讯。除错监视程序提供一个简单的命令集来显示内存地址和寄存器、建立和移除断点, 并且运行你的程序。除错监视器组合这些简单的命令去实现如程序下载, 单步调试等高端的请求。

对于已经具备硬件调试支持单元 DSU 的 S698 处理器来讲, Debug Monitor 并不是必需的。

Sailing/ERC32 Debug monitor 安装:

(1) 进入 mon_sailing 目录;

(2) 修改 Makefile 文件里的内存的配置, 然后保存;

(3) 运行 make all;

(4) 把生成的 .src 文件写到目标板的 PROM。

4.2 界面

4.2.1 主界面

主界面由以下几部分组成:

a) 菜单、工具栏: 所有的功能都可从菜单里找到, 一



图 3 工程向导

点 Exit 按钮结束工程向导。显示如图 4:

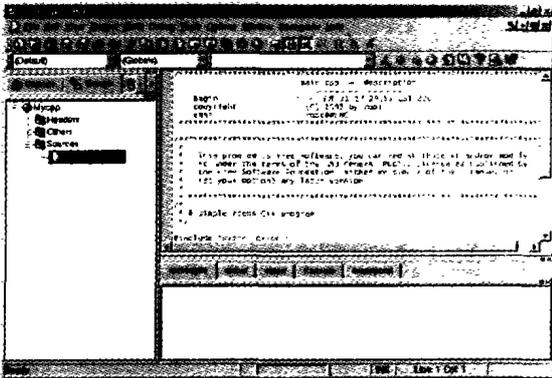


图 4 新工程开始

4.3.2 编译, 连接, 运行

说明: 编译负责把编程语言转变成目标机器代码(*.o 文件); 连接负责把一个或者多个.o 文件组合成一个 elf 格式的可执行文件。

点编译按钮, 对程序进行编译:

Orion 自动调用相应的编译器对其进行编译, 结果应该如图 5 所示。

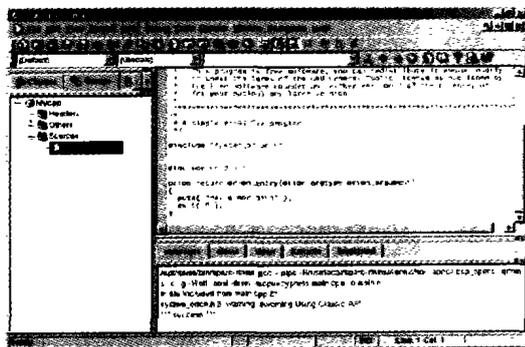


图 5 编译

点连接按钮, Orion 自动调用合适的连接器对编译好的结果进行连接。

最后, 可运行的文件已经生成, 点运行按钮观看运行的结果。

Orion 自动启动合适的模拟器, 并在其之上进行模拟运行, 结果应该如图 6 所示。

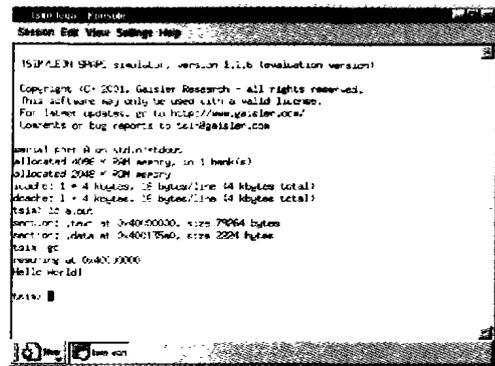


图 6 运行

4.3.3 配置实时系统

说明: Orbita EOS 提供一个可配置的内核, 通过配置内核的参数, 如: 最大任务数, 最大信号量数, 最大消息队列数等参数, 来获得最小的资源开销和最优的整体性能。

点菜单 Project>Options..., 或者按快捷键 F7, 打开 Orion 系统配置界面, 在这里可以选择相应的处理器平台(CPU), 应用程序接口(API), Orion 内核参数。

Orion 通用的选项, 如图 7 所示。

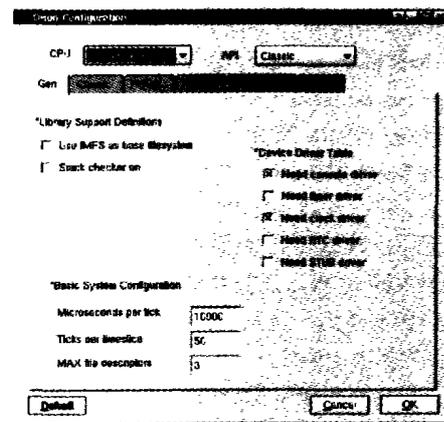


图 7 Orion 通用的选项

如果不清楚某一个选项的用途, 可以先点击右上角的“?”, 再点击该选项, 可以获得该选项的功能介绍。

Orion API 的选项, 图 8 只显示 Classic API 的选项, POSIX, iTRON 的选项与此类似:

Orion Linker 选项, 如图 9 所示。

提供指定各个段的起始地址, 特别是针对 PC386, 在使用 Orion 提供的附件 PC386 Debug monitor 调试应用程序时, 需要将 Code 指定为 00180000。

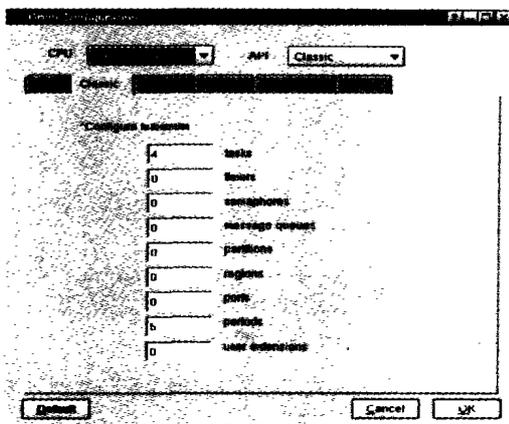


图 8 Orion API 的选项

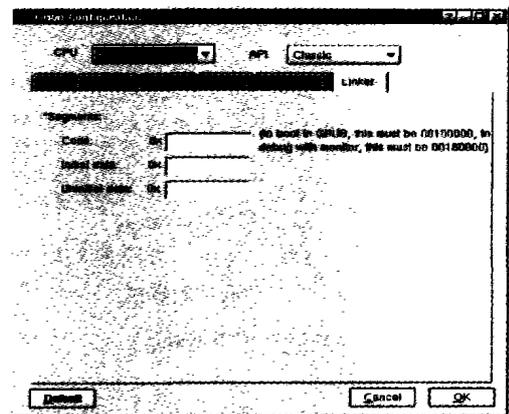


图 9 Orion Linker 选项

4.3.4 调试

说明：如果程序没有出现预期的结果，就需要进行调试。Orion 强大的调试功能允许对程序进行单步跟踪，设置断点，观察变量，察看堆栈等等。

Orion 3.0 支持以下的调试方式：

- (1) 仿真器调试模式，支持 ERC32/Sailing 仿真器；
- (2) Debug monitor 调试模式 (TRIAL version 不支持)；
- (3) DSU 调试模式 (TRIAL version 不支持)。

仿真器调试模式：

说明：如图 10，使用仿真器来模拟 S698 目标板，Orion 和仿真器连接。

Debug monitor 调试模式：

说明：如图 11，Debug Monitor 在目标板上运行，一般



图 10 仿真器调试模式示意图

是烧制在目标板的 ROM 里，系统上电后自动运行 Debug Monitor，Debug Monitor 处理完系统的初始化工作后，打开 COM2 端口，等待 Orion 的连接。

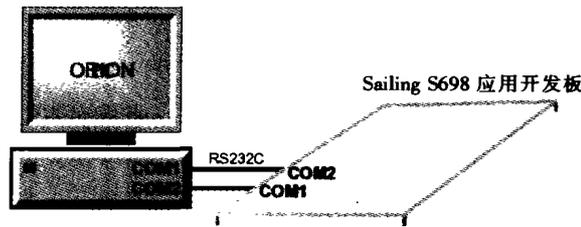


图 11 Debug Monitor 调试模式示意图

说明：如图 12A，可以允许多台 Host PC 连接一台 Target PC，这样就可以做到多人共用一块开发板，需要说明的是：在同一时间只能由一个人使用开发板。使用这种连接，Host PC 可以使用远程控制软件 (例如：VNC) 来控制 Target PC 的 DSUmon 和 Minicom。

图 12B 其实是图 12A 的一个特例，直接使用 Target PC 作为 Host PC，其他的 Host PC 仍然可以如图 12A 所示连接到该 PC。修改工程里的 go_remote 脚本文件时，须指定自己的 IP 地址，或者指定为 localhost。

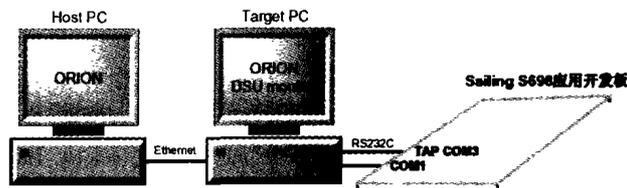


图 12A DSU 调试模式示意图 1

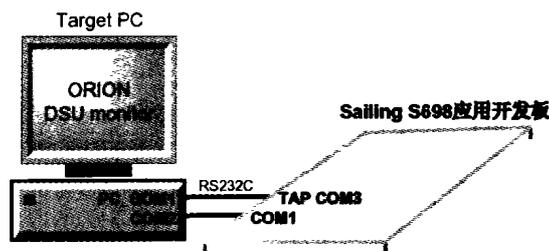


图 12B DSU 调试模式示意图 2

- 使用仿真器调试模式连接时：
参见以下的调试例子。
- 使用 Debug monitor 调试模式时：
 - a) 启动 Sailing S698 目标板并确认 debug monitor 已经在等待连接；
 - b) 确认工程里的脚本文件 go_local 里的串口波特率和串口号正确。参照以下例子修改：

```
file a. out
set remotebaud 38400
```

target extended - remote /dev/ttyS1

load

38400 是 Sailing S698 开发板 UART 和 PC 串口的连接波特率, /dev/ttyS1 表示 PC 的串口 2, 如果通过串口 1 连接到 Sailing S698 开发板, 须修改为 /dev/ttyS0。

c)选 Debug > Start(other...)> Debug with arguments, 填入 -l 或者 -local。

d)等待连接成功并进行调试, 调试方式和使用模拟器调试一样。

• 使用 DSU 调试模式时:

a)启动 Target PC 上的 DSU 监视器(dsumon -i -gdb)并进入等待 GDB 连接状态。

b)确认工程里的脚本文件 go_remote 里的 IP 地址和 Target PC 的 IP 地址一致。参照以下例子修改:

如果采用图 12A 的连接:

file a.out

target extended - remote 192.168.0.31:1235

load

192.168.0.31 表示 Target PC 的 IP 地址, 根据实际情况修改。

如果采用图 12B 的连接:

file a.out

target extended - remote localhost:1235

load

localhost 表示本机, 或者写本机的 IP 地址也可以。

c)选 Debug > Start(other...)> Debug with arguments, 填入 -r 或者 -remote。

d)等待连接成功并进行调试。

例子:以模拟器调试为例,设置好断点,点 Debug 按钮:

Orion 提示将自动启动 Emulator, 请务必等待 Simulator 启动完毕后,再点 OK 按钮。如图 13 所示。

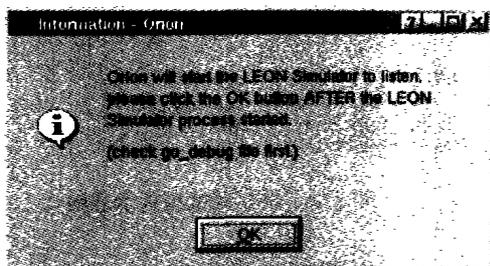


图 13 模拟器调试模式提示

当 Orion 进入调试界面时,如图 14 所示。

- 设置的断点由蓝色变成红色,表示已经找到对应的指令。
- 绿色的指针表示程序当前运行的位置。
- 左边自动切换到观察视图,可以在此添加/观察变量的值,当指令执行后,被改变的值会用红色突出显示
- 信息窗口的 call stack 可以看到函数调用的情况。

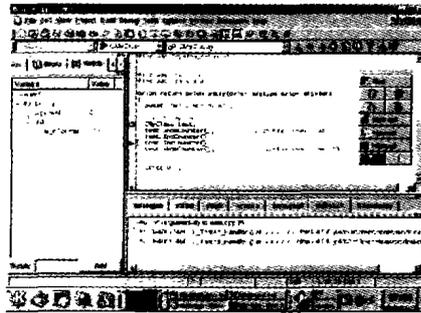


图 14 调试界面

• 信息窗口的 dissembler 可以对当前运行指针位置附近的代码进行反汇编。

• 如果在 Options 里选择了可浮动的调试工具栏,该工具栏显示如图 15:

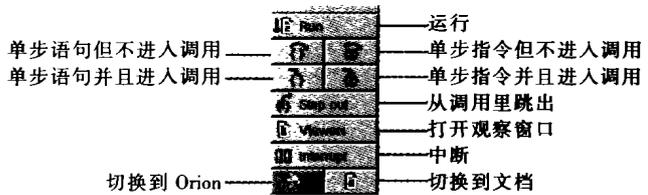


图 15 浮动调试工具栏

调试完毕后点停止按钮返回编辑界面。

4.3.5 生成 ROM Image

说明:生成 ROM Image 是嵌入式应用开发的最后一个步骤,当代码经过调试无误后,就可以生成 ROM 映像文件烧制到目标系统的 ROM 里。

ROM 映像文件一般存在目标系统的 ROM 里,当系统启动后首先被运行,ROM 映像文件负责处理以下的工作:

- 初始化 IU & FPU;
- 通过 ERC32/S698 的控制寄存器,初始化内存参数;
- 把程序解压缩到 ram(使用修改过 LZSS 算法);
- 代码段写保护(可选);
- 设置内存顶部,堆栈指针,然后运行程序。

点菜单 Project>Options..., 或者按快捷键 F7, 打开 Orion 系统配置界面,选 Rom Image 页,如图 16。

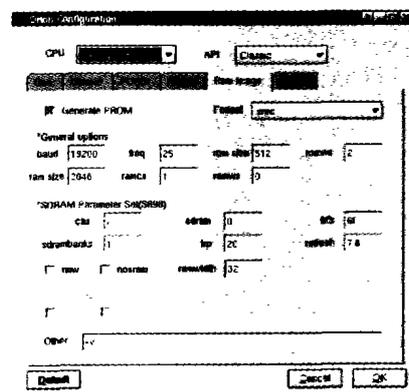


图 16 Rom Image 页 (下转第 45 页)

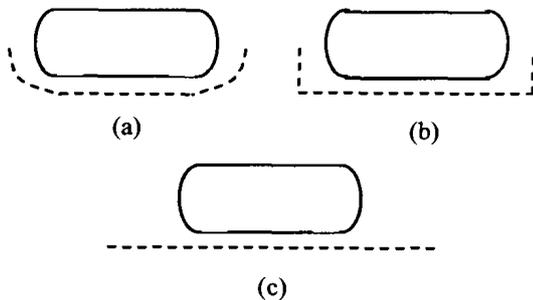


图 4 丝嘴运动轨迹

圆柱(图 4(b))、开放圆柱(图 4(c)),在实际缠绕时,应根据缠绕设备功能、制品形状和精度、效率的要求,依据实际情况选用。

3.2.2 各坐标运动轨迹生成与误差处理

丝嘴运动路线确定后,就可根据缠绕运动的数学模型来确定在各个离散点各个运动坐标的绝对位置。

纤维缠绕的过程是首先按规定的线型在芯模圆周上作螺旋布线,然后按纤维束宽度经过若干次循环布满芯模,因此,若要形成一定的厚度,必然要经过成千上万次循环,所以各坐标运动必须避免累积误差。对于数控缠绕机,其分辨率是一定的,CAD/CAM 系统生成的运动轨迹数据必须以机床的分辨率为单位进行转换,由此造成的舍入误差是造成累积误差的主要原因。纤维缠绕时,一般是走增量坐标,在本系统中通过使往复运动的小车、伸臂、绕丝嘴的正负指令之和为零来消除累积误差。

3.2.3 加工仿真与干涉检查

各坐标轨迹生成后,可以进行丝嘴运动的图形仿真(如图 5 所示)。由于输入参数不正确或由于丝嘴轨迹规划不合理,可能发生丝嘴运动时与芯模或缠绕设备干涉的情况,另外,由于丝嘴都是有一定宽度和高度,也可能发生丝嘴边缘部位出现干涉的情况。所有这些都可以在仿真的过程中及时发现,加以调整,从而避免发生事故。

3.2.4 后置处理

后置处理的主要功能是将各个运动坐标由芯模坐标系转化到机床坐标系中,并进行速度的优化,自动生成数控代码。在本系统中,采用三种速度优化方式,分别是最小加工时间、恒出纱速度和恒芯模转速。在最小加工时间模

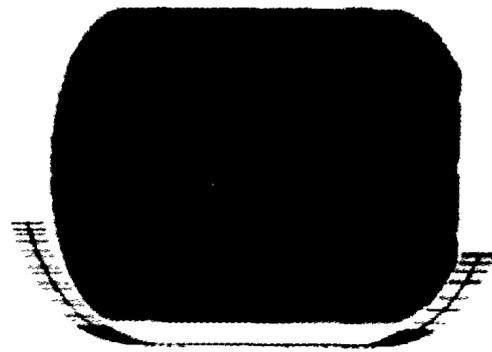


图 5 缠绕仿真与干涉检查

式下,各轴速度在不超限的前提下,构成合成缠绕速度的速度最高的运动轴将以设备允许的最高的速度运行,效率最高;在恒出纱速度模式下,减小了纤维张力的变化,保证了较高的纤维张力控制精度;恒芯模转速模式比较适合于大型制品的缠绕,将减小由于主轴不断加减速而给主电机和传动机构造成的冲击。自动生成的符合数控代码格式的 NC 加工程序,将通过通讯接口传输至相应的数控缠绕设备。

4 结束语

纤维缠绕成型 CAD/CAM 系统的开发,可大大减少人工繁琐及重复的计算工作,提供多种设计方案,缩短设计周期,保证纤维按最佳线型进行缠绕,提高产品质量和成品率。文中将 CAD/CAM 技术应用于纤维缠绕成型工艺,在分析纤维缠绕规律和各坐标运动关系的基础上,对系统软件的组成和各模块的功能进行了分析和研究。基于此思想,哈工大数控缠绕机研究所开发了纤维缠绕 CAD/CAM 软件 WINDSOFT,在实际应用过程中,取得了良好的效果。

参考文献

- 1 孔庆宝. 纤维缠绕技术进入新的高速发展阶段. 纤维复合材料, 1998(3): 35.
- 2 <http://wind.hit.edu.cn>.
- 3 Jae-Sung Park, Chang-Sun Hong, Chun-Sun Hong, Chun-Gon Kim, Cheol-Ung Kim. Analysis of Filament Wound Composite Structures Considering the Change of Winding Angles Through the Thickness Direction. Composite Structures:2002(55): 63~71.

(上接第 38 页)可以选择在生成目标代码的时候,是否同时产生 ROM 映像文件;ROM 映像文件的多种格式如表 1,分别对应不同的文件名,可以根据 prom 编程器的要求选择一种:

参考文献

- 1 www.kdevelop.org. KDevelop User Manual.
- 2 www.gnu.org. GCC/GDB User Manual.
- 3 欧比特公司. Orbita EOS Classic API Guide, 2003.
- 4 欧比特公司. S698 处理器应用开发系统用户手册, 2003.

表 1 PROM 格式

类型	Prom 文件名	说明
Srec	Orion.srec	SRecord 格式
Ihex	Orion.ihex	Intel HEX 格式
Binary	Orion.bin	二进制格式
Elf 32-sparc	Orion.elf	Elf 32 格式
Coff-sparc	Orion.coff	Coff 格式