

HNC-21T 控制系统的 VERICUT 控制文件开发

撰文 / 天津机电职业技术学院产学研中心 赵华 陈甫

对华中世纪星 HNC-21T 数控车系统进行分析, 在 VERICUT 仿真软件中配置对应的控制文件, 通过 VERICUT 交互界面对 HNC-21T 系统有代表性的指令和功能配置进行探讨, 介绍了 VERICUT 配置控制文件的一般步骤和方法, 通过对 NC 子程序、VERICUT 宏、数控宏程序等多种方法的综合运用, 对 G80 外径车削单一循环指令进行配置。通过一个典型零件的虚拟加工, 验证了所配置的控制文件的正确性。

一、引言

VERICUT 软件是一款美国 CGTech 公司开发的数控加工仿真和优化平台, 可以真实地模拟多种加工设备的数控加工过程, 能够进行 NC 程序优化, 检查过切、欠切, 防止机床碰撞、超行程等错误。VERICUT 为用户提供了大量仿真机床和控制系统, 但对于一些非标数控设备和国际市场占有率低的设备和控制系统, 需要用户根据自己的使用需求进行自定义。近年来, 国产数控产品日益崛起, 但是 VERICUT 库中几乎没有国产数控设备及系统样本。本文针对国内广泛使用的华中世纪星 HNC-21T 数控车床系统进行配置, 介绍了 VERICUT 数控系统配置的一般步骤和方法, 并对其中的关键技术进行研究。

二、VERICUT 控制系统

在 VERICUT 中, 一个完整的仿真项目由项目文件 (*.vcproject)、机床文件 (*.mch)、控制文件 (*.ctl)、刀具文件 (*.tls) 及模型文件 (*.stl) 组成。其中机床文件包括各种运动轴的配置、夹具和毛坯等, 即数控机床硬件; 控制文件包含各种指令代码及符号配置, 即数控机床控制系统。VERICUT 控制系统是虚拟的控制系统, 任务是向 VERICUT 软件解释机床代码, 并控制虚拟机床运动。

用户可以根据自己的需求直接调用 VERICUT 样本库中已有的控制系统文件, 也可以选用 VERICUT 提供的宏在交互界面中建立新的控制文件。VERICUT 的宏是开发者为了方便用户使用, 开发的一些程序块, 供用户选择使用, 可以用来实现机床的某些动作, 也可以配置系统指令。VERICUT 控制系统与实际控制系统十分相似, 因此用户在建立控制系统时可以按照系统编程说明书对控制系统的功能和指令逐一进行配置, 也可以仅对用户需要的功能进行选择性的配置。

自定义控制系统文件, 可以选择某一 VERICUT 库内的控制文件作为模板, 在模板的基础上进行修改。一般, VERICUT 控制系统的制定包括以下几个方面: Word Format (字格式的定义), G-Code Processing (代码定义), Control Settings (控制系统设置), Adv. Options (高级选项)。VERICUT 可以支持一般的标准代码, 但是对一些非标准代码和格式需要进行定义。Configuration (配置) > Word Format (字格式), 打开 Word Format window (字格式定义窗口), 定义指令字格式。Configuration (配置) > G-Code Processing (代码定义), 定义每一个字的功能。下面选取 HNC-21T 系统具有代表性的指令和功能进行配置。

三、编程方式设置

1. 绝对值编程与相对值编程

格式：G90/ G91。说明：G90 是绝对值编程，指令后面的 X、Z 表示 X 轴、Z 轴的坐标值；G91 是相对值编程，用 U、W 或 G91 指令后面的 X、Z 表示 X 轴、Z 轴的增量值，G90 为缺省值。Configuration (配置) > Word Format (字格式)，打开 Word Format window (字格式定义窗口)，定义字 U、W 如表 1 所示。

表 1 U、W 字格式的定义

Name 名称	Type 类型	Sub Type 子类型	Inch Method 英制类型	Inch Format 英制格式	Metric Method 公制类型	Multiple 系数
U	Macro	Numeric	Decimal	3.4	Decimal	No
W	Macro	Numeric	Decimal	3.4	Decimal	No

Configuration (配置) > G-Code Processing (代码定义) > Word/Address，对绝对值编程 G90 与相对值编程 G91 进行设置如图 1 所示。



图 1 G90/ G91 G-Code Processing 代码设置

2. 直径方式编程和半径方式编程

格式：G36/G37。说明：G36 为直径编程，G37 为半径编程，G36 为缺省值，机床出厂一般设为直径编程。首先通过程序测试系统模板的默认编程方式，经过测试调用的该模板的默认编程方式是直径方式编程，因此 G36 直径编程和 G37 半径编程的 G 代码配置如图 2 所示。

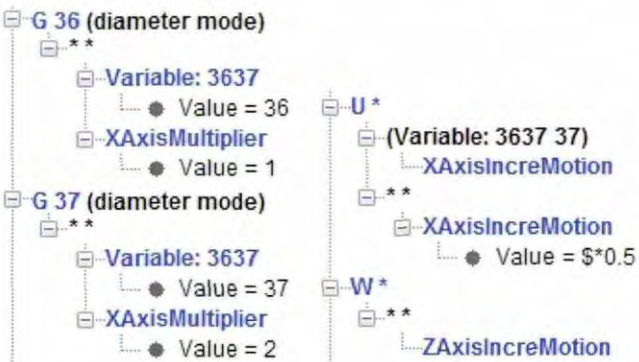


图 2 G36/G37 G-Code Processing 代码设置

四、宏程序功能

HNC-21 为用户配备了宏程序功能，用户可以使用变量进行算数运算、逻辑运算和函数的混合运算，此外宏程序还提供了循环语句、分支语句和子程序调用语句。以 IF、ELSE、ENDIF 语句为例，格式定义如表 2 所示。

表 2 宏程序语句格式定义

Name 名称	Type 类型	Sub Type 子类型	Inch Method 英制类型	Metric Method 公制类型	Multiple 系数
IF	Macro (宏)	Numeric	Decimal (十进位)	Decimal (十进位)	No
ELSE	Macro	None			
ENDIF	Macro	None			

Configuration (配置) > G-Code Processing (代码定义) > Word/Address > Branching，对语句的功能进行设置，如图 3 所示。



图 3 部分宏语句 G-Code Processing 设置

五、车削循环

切削循环通常是用一个含 G 代码的程序段完成多个程序段指令的加工操作，使程序得以简化。VERICUT 提供了丰富的宏，可以对一般的循环命令通过 G-Code Processing 进行配置，但是也有些特殊的或者不经常使用的循环指令无法用宏来实现，这时可以使用子程序的方法进行定义。本文以 G80 为例说明 VERICUT 中通过子程序的方法对循环指令进行配置的方法。

格式：G80 X_Z_ I F_。说明：G80 为外圆轮廓内外径单一车削循环，循环路线如图 4 所示；X、Z 为绝对值编程时切削终点 C 在工件坐标系下的坐标值，如果是增量坐标编程，则用 U、W 表示；I 为切削起点 B 与切削终点 C 的半径差。

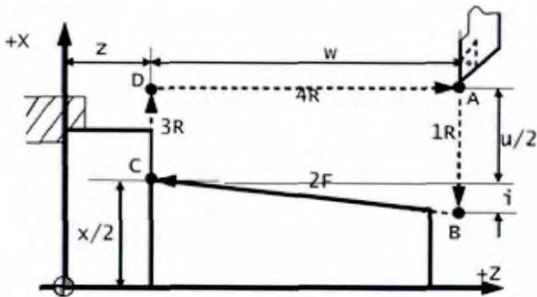


图 4 G80 循环指令路线图

首先通过 Project > Variables 定义子程序中所需要的变量，变量名称及意义如表 3 所示。

表 3 变量名称及意义

名称	含义	初始值
3637	值为 36 为直径编程，值为 37 为半径编程	36
8000	循环类型	0
8001	G80 指令后的 X 坐标值，即循环终点 C 的 X 坐标值	0
8002	G80 指令后的 Z 坐标值，即循环终点 C 的 Z 坐标值	0
8003	G80 指令执行前的 X 坐标值，即循环起点 A 的 X 坐标值	0
8004	G80 指令执行前的 Z 坐标值，即循环起点 A 的 Z 坐标值	0
8005	G80 指令中的 I 值	0
8006	直径编程时值为 1，半径编程时值为 2	1
8007	绝对值编程下，循环终点 C 的 X 坐标值	0
8008	绝对值编程下，循环终点 C 的 Z 坐标值	0
8009	判断当下执行的 NC 语句是否在 G80 子程序内部	0
8010	默认值为 0，执行过 G80 指令后值为 80	0
8011	执行 G80 子程序前的编程方式，值为 91 为增量编程，90 为绝对编程	0

Configuration（配置）> G-Code Processing（代码定义），对 G80 指令相关的字进行设置，如图 5 所示。其中以 G0 为代表说明所有可以注销 G80 的指令设置方法。

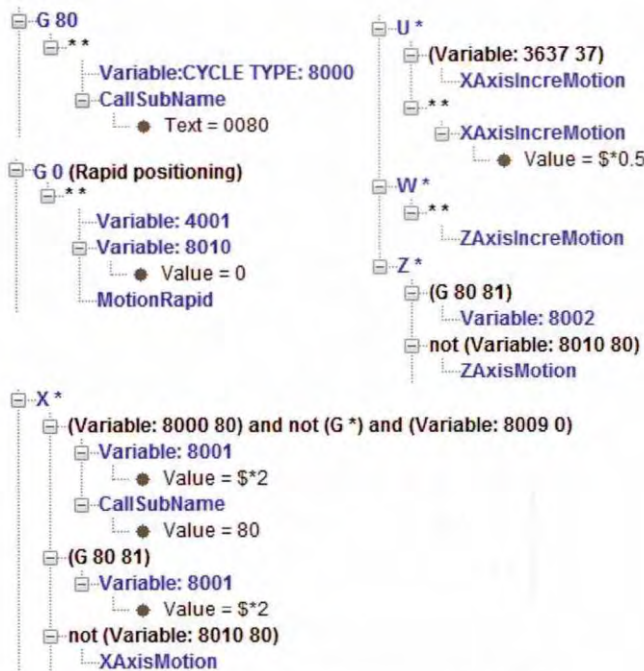


图 5 G80 相关字的 G-Code Processing 设置

宏程序采用直径编程，子程序主体及注释如下。

```

O0081
#8010=0
#8009=1
#8003=#5041 （起点位置 X 坐标值 #5041 赋值给 #8003）
#8004=#5042 （起点位置 Z 坐标值 #5042 赋值给 #8004）
IF #3637 EQ 37
    #8006=2（半径方式编程）
ELSE
    #8006=1（直径方式编程）
ENDIF
IF #4003 EQ 91（如果 X 值是增量方式 G91）
    #8007=#8001/#8006+#8003（增量 X 值转化为绝对直径值，赋值给 #8007）
    #8011=91（把执行 G80 子程序之前的编程方式赋值给 #8011）
ELSE
    #8007=#8001/#8006（X 值转化为直径值，赋值给 #8007）
    #8011=90（把执行 G80 子程序之前的编程方式赋值给 #8011）
ENDIF
IF #4003 EQ 91（如果 Z 值是增量方式 G91）
    #8008=#8002+#8004（增量 Z 值转化为绝对值，赋值给 #8008）
ELSE
    #8008=#8002
ENDIF
G90 （将编程方式设置为绝对坐标）
CGTECH_MACRO "MotionRapid" （快进）
X[#8007+#8005*2] （从 A 点移动到 B 点）
CGTECH_MACRO "MotionLinear" （进给）
X[#8007] Z[#8008]（从 B 点移动到 C 点）
CGTECH_MACRO "MotionRapid"
X[#8003/#8006]（从 C 点移动到 D 点）
Z[#8004]（返回起点）
#8009=0
#8010=80（设置标记，执行 G80 直到被其他 G 代码注销）
    
```

基于 CATIA 超级副本的交互式参数化建模

撰文 / 中航工业贵州贵航飞机设计研究所 姚旭 赵荣

本文主要叙述了一种基于 CATIA 超级副本的交互式参数化建模方法，其主要内容是通过人机对话的方式，在可视化交互的界面中进行设计输入，软件自动对用户输入的尺寸、几何元素等类型参数进行处理，得到完整模型的建模方法。通过该方法，可以免去枯燥、乏味的建模过程，极大地提高设计效率。

一、引言

对于大型复杂产品，设计过程中经常会遇到“相似零件”的重复建模问题。所谓“相似零件”就是构造特征相同，但参数不同，例如飞机结构中机身的钣金隔框、长桁，机翼的肋等零件，其构造特征相同，但构造参数随着外形及站位变化而变化，零件也各不相同。大型复杂产品一般会有大量的“相似零件”，“相似零件”必须进行单独建模，也就是重复建模。建模时除输入的数据不同外，其余的操作都是相同的，重复操作的比重很大，这种重复操作不仅枯燥乏味，而且效率低。CATIA 软件提供了一种叫做“超

级副本”的功能，使用超级副本功能，可剔除上述重复操作，大大提高设计效率。

二、用 CATIA 软件中的超级副本建模

1. 超级副本功能实质

CATIA 的超级副本功能是参数化设计技术的拓展，实质是参数化和“宏”的有机结合。即在交互式环境中输入“相似零件”的参数，“宏”调用输入的参数，批处理建模操作过程，自动生成零件模型。

CATIA 软件中的超级副本功能，由于介绍该功能的资

#4003=#8011（恢复编程方式）
M99

六、结语

数控机床正向着多功能、多主轴和多通道等方向发展，数控仿真技术日益重要，VERICUT 越来越得到广大机床用户的认可。本文通过对华中世纪星 HNC-21T 数控车系统部分指令和功能的配置，介绍了在 VERICUT 中自定义数控控制文件的一般步骤和方法，并通过对 VERICUT 宏、子程序等方法的应用，配置循环指令 G80。通过对一个能够覆盖本文所有配置功能的零件进行虚拟加工，如图 6 所

示，验证了配置的正确性。

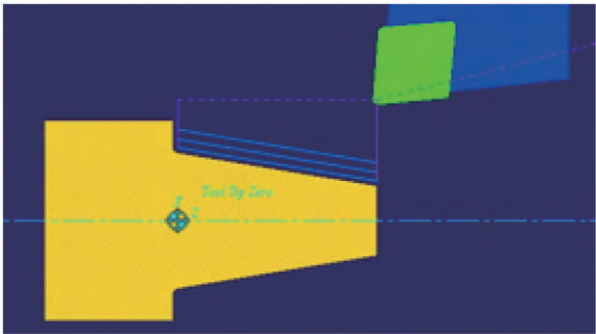


图 6 虚拟加工实例