件处理文件中加入此类事件的子程序。

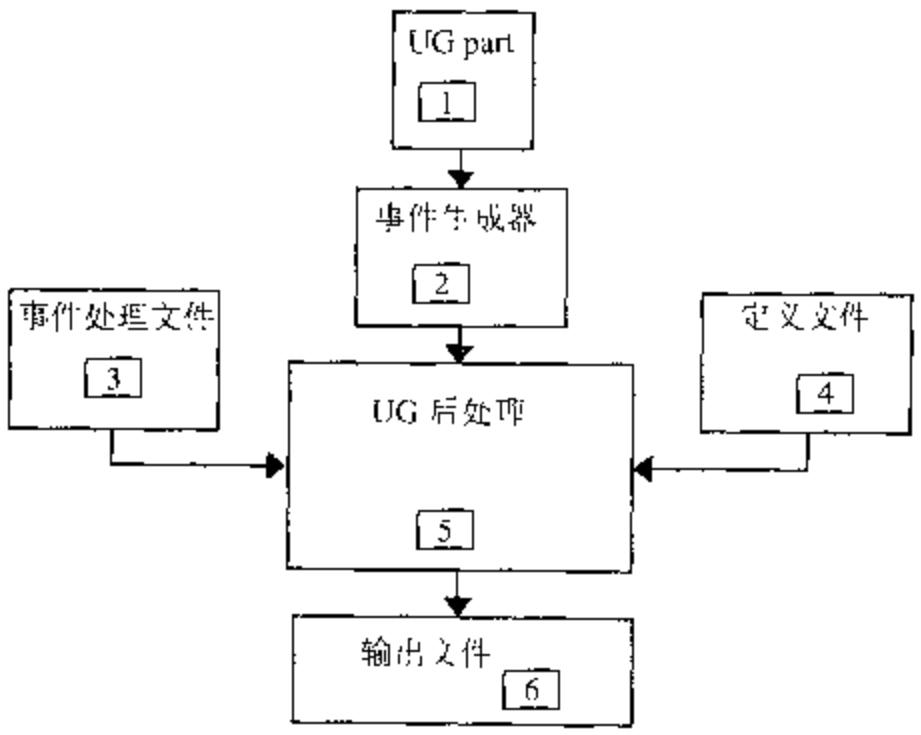


图 9-1 事件处理文件

事件对应的子程序名必须与事件生成器产生的事件名相同。比如：start of program 事件的子程序名是 MOM_start_of_program；tool change 事件的子程序名是 MOM_tool_change。

事件中的相关参数作为全局变量传给事件处理文件。在附录 B 中列出了典型事件的说明和相关变量。

练习 9-1：修改事件处理文件

在这个练习里对以前产生的事件处理文件作多处修改。

第 1 步 修改***_test_post.tcl 文件中的 start_of_program 子程序。

- 双点打开文件***_test_post.tcl。
- 修改 start_of_program 子程序，输出“%4711”，不带 N 号，
- 加一行不带“N”的程序头。
- 输出：mom_date；mom_logname；mom_part_name。
- 加一条指令打开程序行“N”输出开关，
- 另输出一行：“G40 G80 G17”。
- 保存文件。
- 处理文件 pbt_mill_test.prt 中 T12345-A 下的操作，检验输出结果。

第 2 步 修改***_test_post.tcl 文件中的 start_of_path 子程序。

- 修改子程序 start_of_path 输出：

```
mom_operation_type
mom_path_name
```

```
mom_tool_name
mom_tool_type
mom_tool_number
```

- 保存文件。
- 处理文件 pbt_mill_test.prt 中 T12345-A 下的操作，检验输出结果。

第 3 步　修改***_test_post.tcl 文件，输出 G11 作为直线插补，G12 作为顺时针圆弧插补，G13 作为逆时针圆弧插补。

- 在***_test_post.tcl 文件中找到：

```
set mom_sys_linear_code  "1"
```

把 1 改成 11。

- 在同一文件中找到：

```
set mom_sys_circle_code(CIW)  "2"
```

把 2 改成 12。

- 在同一文件中找到：

```
set mom_sys_circle_code(CCLW)  "3"
```

把 3 改成 13。

- 保存文件。
- 处理文件 pbt_mill_test.prt 中 T12345-A 下的操作，检验输出结果。

第 4 步　把 tapping（攻丝）的冷却液代码改成 23。

- 在***_test_post.tcl 文件中找到：

```
set mom_sys_coolant_code(ON)  "8"
```

把 8 改成 23。

- 保存文件。
- 打开冷却液事件，设为 tap。
- 处理文件 pbt_mill_test.prt 中 T12345-A 下的操作，检验输出结果。

第 5 步　修改***_test_post.tcl 文件，把圆弧插补从整圆改成象限。

- 在***_test_post.tcl 文件中找到：

```
set mom_kin_arc_output_mode  "FULL_CIRCLE"
```

把 FULL_CIRCLE 改成 QUADRANT。

- 保存文件。
- 打开冷却液事件，设为 tap。
- 处理文件 pbt_mill_test.prt 中 T12345-A 下的操作，检验输出结果。

练习结束。

9.1.2 定义文件

定义文件包含了机床/控制系统的格式定义信息（如图 9-2 所示）。

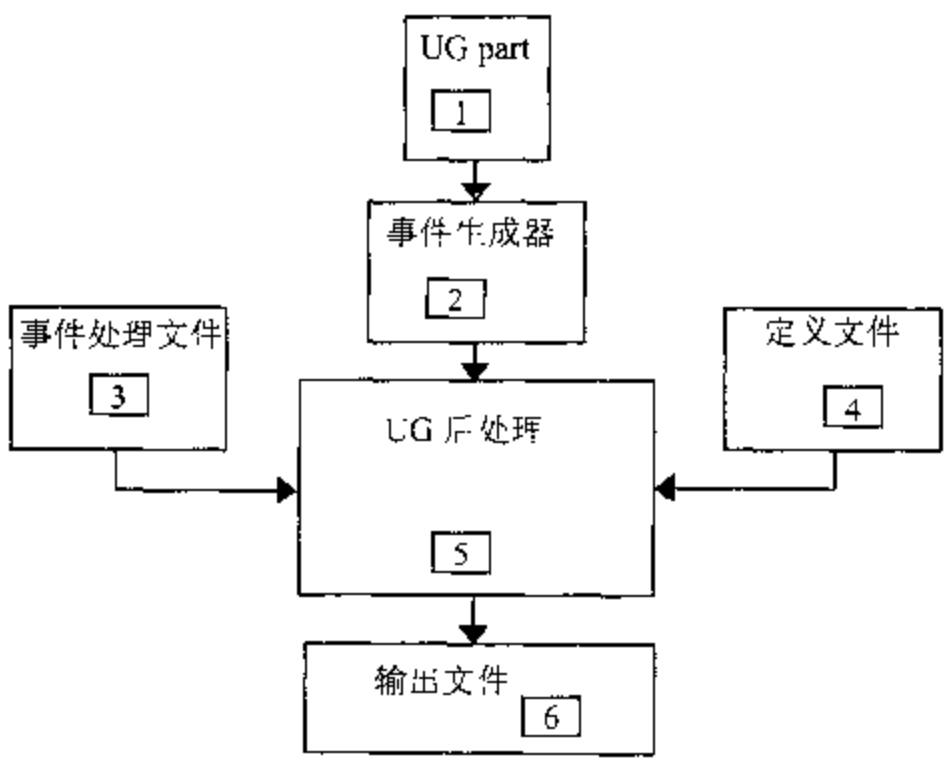


图 9-2 定义文件

大多数 NC/CNC 机床用字地址来控制机床来完成各种功能。比如：字地址 X、Y、Z 用来储存直线插补结束点的 X、Y、Z 的坐标值。NC/CNC 程序中每一条改变机床状态的指令都是由改变字地址的状态实现的。UG/Post 就是用定义文件中的信息来决定指令输出的格式，进而驱动机床运动的。

定义文件中的信息必须与机床的属性相一致，比如换刀、主轴转速和行程极限等。再比如机床使用的字地址：直线插补的 X、Y、Z；主轴转速 S 和切削进给 F 等。还有字地址的属性：数据格式、最大、最小值。还有由一系列字地址组合起来的完成一个完整机床动作的程序行的定义：比如直线插补语句可以定义成 G01 X [X 值] Y [Y 值] Z [Z 值]。

附录 A 是一个定义文件的例子。

练习 9-2：修改定义文件

在这个练习里对一个现成的定义文件作多处修改。

第 1 步 修改***_test_post.def，在字地址之间插入一个空格。

- 双点文件***_test_post.def。
- 修改这个文件，使输出的 NC 程序的两个字地址之间保留两个空格。
- 改变 WORD_SEPARATOR 的值。
- 保存文件。
- 将文件 pbt_mill_test.prt 中 T12345-A 下的操作作后处理输出，检验结果。

- 再修改定义文件，使字地址之间保留一个空格。
- 再改变 WORD_SEPARATOR 的值。
- 保存文件。
- 将文件 pbt_mill_test.prt 中 T12345-A 下的操作作后处理输出，检验结果。

第 2 步　修改***_test_post.def，改变输出序号的格式。

- 打开文件***_test_post.def。
- 把序号 sequence number 定义为从 1 开始，以 1 递增。
- 保存文件。
- 将文件 pbt_mill_test.prt 中 T12345-A 下的操作作后处理输出，检验结果。
- 再修改文件，把序号 sequence number 定义为从 5 开始，以 10 递增。
- 保存文件。
- 将文件 pbt_mill_test.prt 中 T12345-A 下的操作作后处理输出，检验结果。

第 3 步　修改***_test_post.def，改变输出序号的格式。

- 打开文件***_test_post.def。
- 把序号 sequence number 定义为从 10 开始，以 10 递增，没有前 0（格式定义参考附录 B）。
- 保存文件。
- 将文件 pbt_mill_test.prt 中 T12345-A 下的操作作后处理输出，检验结果。
- 再修改文件，把序号 sequence number 定义为从 10 开始，以 10 递增，输出前 0（共 4 位数）。
- 保存文件。
- 将文件 pbt_mill_test.prt 中 T12345-A 下的操作作后处理输出，检验结果。

第 4 步　修改***_test_post.def，改变坐标值的格式。

- 打开文件***_test_post.def。
- 修改坐标输出格式，输出 5 位数，3 位小数，没有前 0、后 0 和小数点。
- 保存文件。
- 将文件 pbt_mill_test.prt 中 T12345-A 下的操作作后处理输出，检验结果。
- 修改坐标输出格式，输出 4 位数，4 位小数，有前 0、后 0，没有小数点。
- 保存文件。
- 将文件 pbt_mill_test.prt 中 T12345-A 下的操作作后处理输出，检验结果。

第 5 步　修改***_test_post.def，改变字地址输出。

- 打开文件***_test_post.def。
- 把输出的 Y 变为 X，X 变为 Y。
- 保存文件。
- 将文件 pbt_mill_test.prt 中 T12345-A 下的操作作后处理输出，检验结果。
- 再修改文件，在 first move 中加入 Z 坐标值。
- 保存文件。
- 将文件 pbt_mill_test.prt 中 T12345-A 下的操作作后处理输出，检验结果。

第 6 步　修改钻孔循环的输出。

- 将钻孔循环的输出改为：

  G81 X Y E R

  X Y 是钻孔开始点位置；

  E 是孔的深度（一般用 Z 表示）；

  R 是快速定位点。
- 保存文件。
- 将文件 pbt_mill_test.prt 中 T12345-A 下的操作作后处理输出，检验结果。

第 7 步　修改圆弧插补的输出。

- 修改圆弧插补，输出圆心坐标。
- 保存文件。
- 将文件 pbt_mill_test.prt 中 T12345-A 下的操作作后处理输出，检验结果。

第 8 步　修改快速移动。

- 把 G00 改成 G01，再加上 R。

  G01 X Y Z R
- 保存文件。
- 将文件 pbt_mill_test.prt 中 T12345-A 下的操作作后处理输出，检验结果。

第 9 步　修改快速移动。

- 在所有快速移动行后加 F0。

  G00 X Y Z F0
- 保存文件。
- 将文件 pbt_mill_test.prt 中 T12345-A 下的操作作后处理输出，检验结果。

练习结束。

9.2　机床运动学

机床运动学是讲机床各轴的运动。机床运动变量定义事件生成器所涉及的机床运动信息，UG/Post 就是用这些机床运动变量作处理的。事件处理文件定义机床运动变量，机床运动变量定义如何把刀轨坐标 X、Y、Z、I、J、K 转换成机床坐标 X、Y、Z、A、B。机床运动变量还定义机床的基本类型，如车、铣或线切割。

9.2.1　机床类型

机床运动变量 mom_kin_machine_type 定义的机床基本类型有：

- 3_Axis_mill；
- 4_Axis_head；

- 4_Axis_table;
- 5_Axis_dual_table;
- 5_Axis_dual_head;
- 5_Axis_head_table;
- 2_Axis_wedm;
- 4_Axis_wedm。

当机床类型设为车时，所有刀轨坐标 X、Y、Z 就对应转化为车床运动数据 X、Z。

当机床类型设为铣时，所有刀轨坐标 X、Y、Z、I、J、K 就对应转化为铣床运动数据 X、Y、Z、A（第4轴）、B（第5轴）。

机床类型还定义机床运动轴的数量。每一旋转轴都有下列要素：

- 旋转平面；
- 行程极限；
- 旋转方向；
- 零点位置。

9.2.2 圆弧插补

机床运动变量 mom_kin_arc_output_mode 定义圆弧插补输出的数据和行数。变量的相关参数决定圆弧输出类型：

- Full_Circle——可以输出整圆；
- Quadrant——圆弧按象限分段输出；
- Linear Arcs——圆弧分成一段段直线输出。

机床运动变量 mom_kin_arc_valid_plane 定义事件生成器在哪个平面生成圆弧插补。有如下参数：

- XYZ——在三个主平面生成圆弧插补；
- XY——在 XY 平面生成圆弧插补；
- YZ——在 YZ 平面生成圆弧插补；
- ZX——在 ZX 平面生成圆弧插补。

9.3 高级机床运动学

对于 5 轴机床，机床运动变量 mom_kin_machine_type 有以下参数来定义机床类型（如图 9-3）：

- 5_axis_dual_table;
- 5_axis_head_table;
- 5_axis_dual_head。

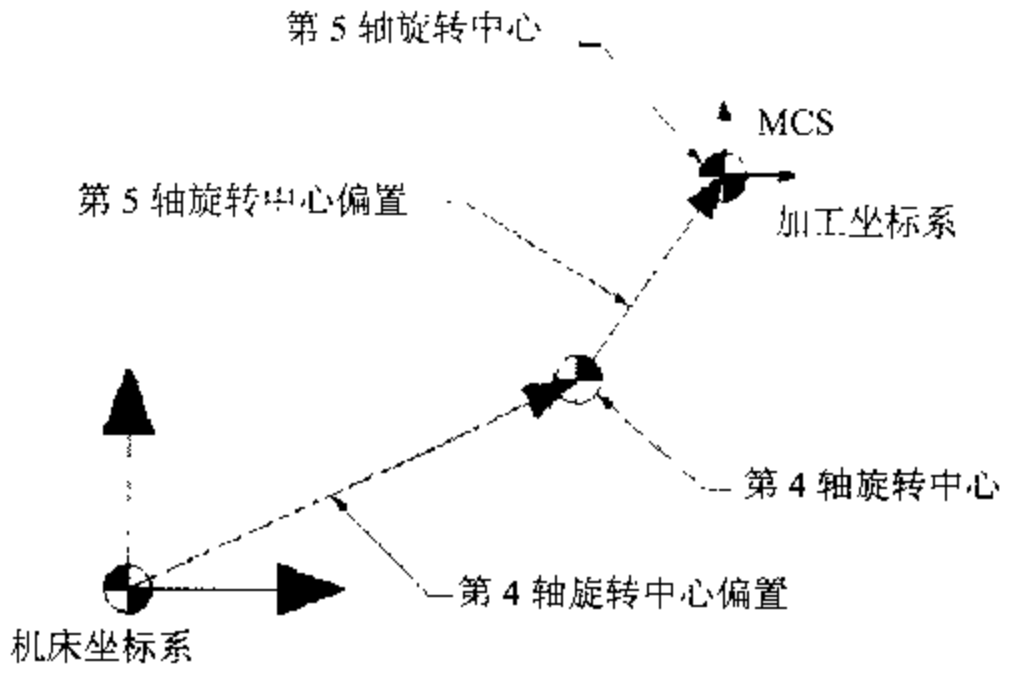
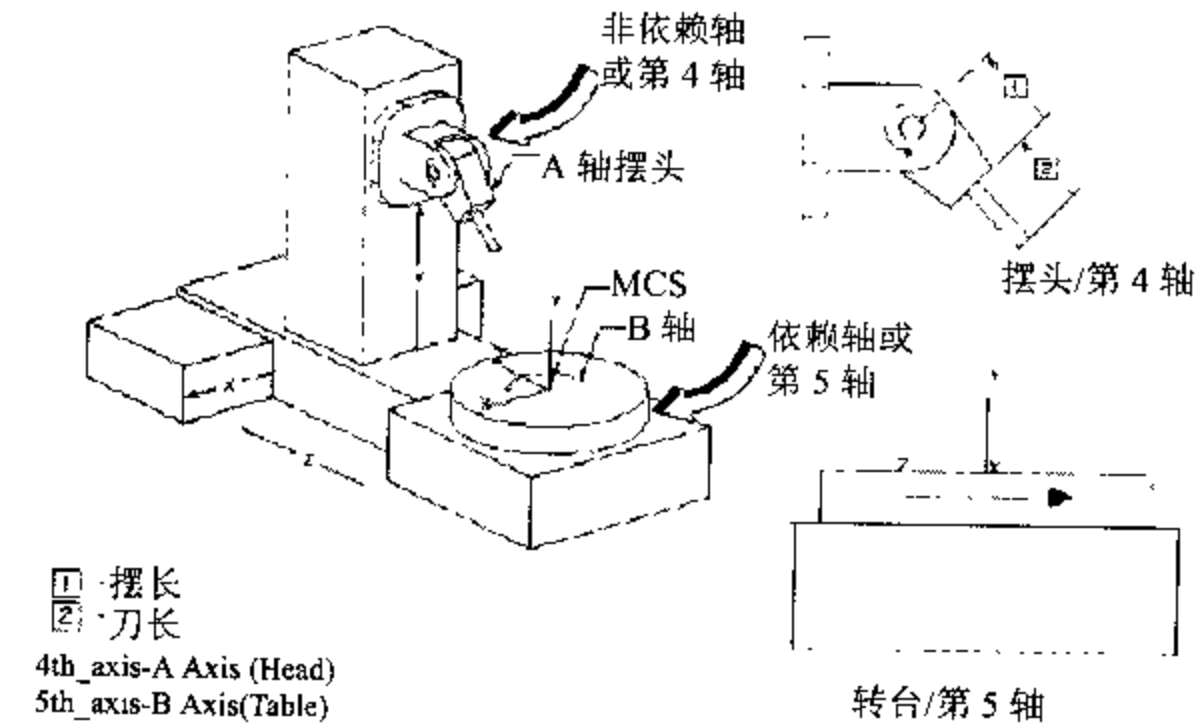


图 9-3　第 4 轴与第 5 轴坐标系关系图

9.3.1　旋转轴指定

5 轴机床有依赖轴和非依赖轴。当另一个轴旋转时，这个轴不改变旋转方向和旋转平面，这个轴就是非依赖轴，也是第 4 轴；当另一个轴旋转时，这个轴改变旋转方向和旋转平面，这个轴就是依赖轴，也是第 5 轴。

对于一个 5_axis_head_table 机床，摆头永远是非依赖轴，也就是第 4 轴，而转台是第 5 轴。在图 9-4 中 C 轴不能定义为第 4 轴。



变量	值
mom_kin_5th_axis_center_offset(0)	0
mom_kin_5th_axis_center_offset(0)	0
mom_kin_5th_axis_center_offset(0)	0

图 9-4　摆台-转台五轴机床

1. 第 4 轴旋转中心偏置

数组变量 mom_kin_4th_axis_center_offset 中含有机床坐标零点到第 4 轴旋转中心的偏差值。这些值必须设置正确，以使 UG/Post 能精确的将零件坐标系转化为机床坐标系。

- mom_kin_4th_axis_center_offset (0)　　X 方向偏差
- mom_kin_4th_axis_center_offset (1)　　Y 方向偏差
- mom_kin_4th_axis_center_offset (2)　　Z 方向偏差

2. 第 5 轴旋转中心偏置

数组变量 mom_kin_5th_axis_center_offset 中含有第 5 轴旋转中心到第 4 轴旋转中心的偏差值。矢量的计算方法是：当第 5 轴的旋转轴与第 4 轴的旋转轴不相交时，两轴都回到零位，在第 4 轴的旋转平面里从第 4 轴旋转轴垂直地指向第 5 旋转轴。一般情况只有 X、Y、Z 中一个方向的偏差。

- mom_kin 5th_axis_center_offset (0)　　X 方向偏差
- mom_kin_5th_axis_center_offset (1)　　Y 方向偏差
- mom_kin_5th_axis_center_offset (2)　　Z 方向偏差

9.3.2 轴的转向（标准/反向）

轴的转向变量 mom_kin_4th (5th) axis_rotation 有 standard 和 reverse 两中参数，控制如何把表示刀轴方向的 I、J、K 数据转换为转角。

- mom_kin_4th_axis_rotation　　Standard
- mom_kin_4th_axis_rotation　　Reverse
- mom_kin_5th_axis_rotation　　Standard
- mom_kin_5th_axis_rotation　　Reverse

用下面的规则来定义转向：

- 大多数 CNC 机床用 Standard，表示转轴或摆轴是以顺时针方向转向大角度。不能确定时，先用 Standard 来看一下输出是否正确。
- 在 ZX 平面，沿 Y 轴从正向负看，如果顺时针是转向大角度，就是 Standard。
- 在 YZ 平面，沿 X 轴从负向正看，如果顺时针是转向大角度，就是 Standard。

每一个转轴或摆轴都要确定转向，或是标准，或是反向。

如图 9-5 的一个双摆轴机床，可以把刀轴转到（0,0,1）的位置。当 A 轴定义成 Standard 时，需转 90°；当定义成 Reverse 时，需转-90°（如图 9-5 所示）。

如图 9-6 所示是这个机床的运动参数定义。

对于这种类型的机床，从机床零点到第 4 轴旋转中心的偏差必须设成 0。

1. 零点偏差

当机床的两个转轴都转到零位时，刀轴在机床坐标系的 Z 轴方向。

有些机床（双摆头机床或摆头+转台机床）的零点在极限位置。

这些值可以定义在机床运动变量 mom_kin_4th_axis_ zero_position 和 mom_kin_5th_axis

_zero_position 中。

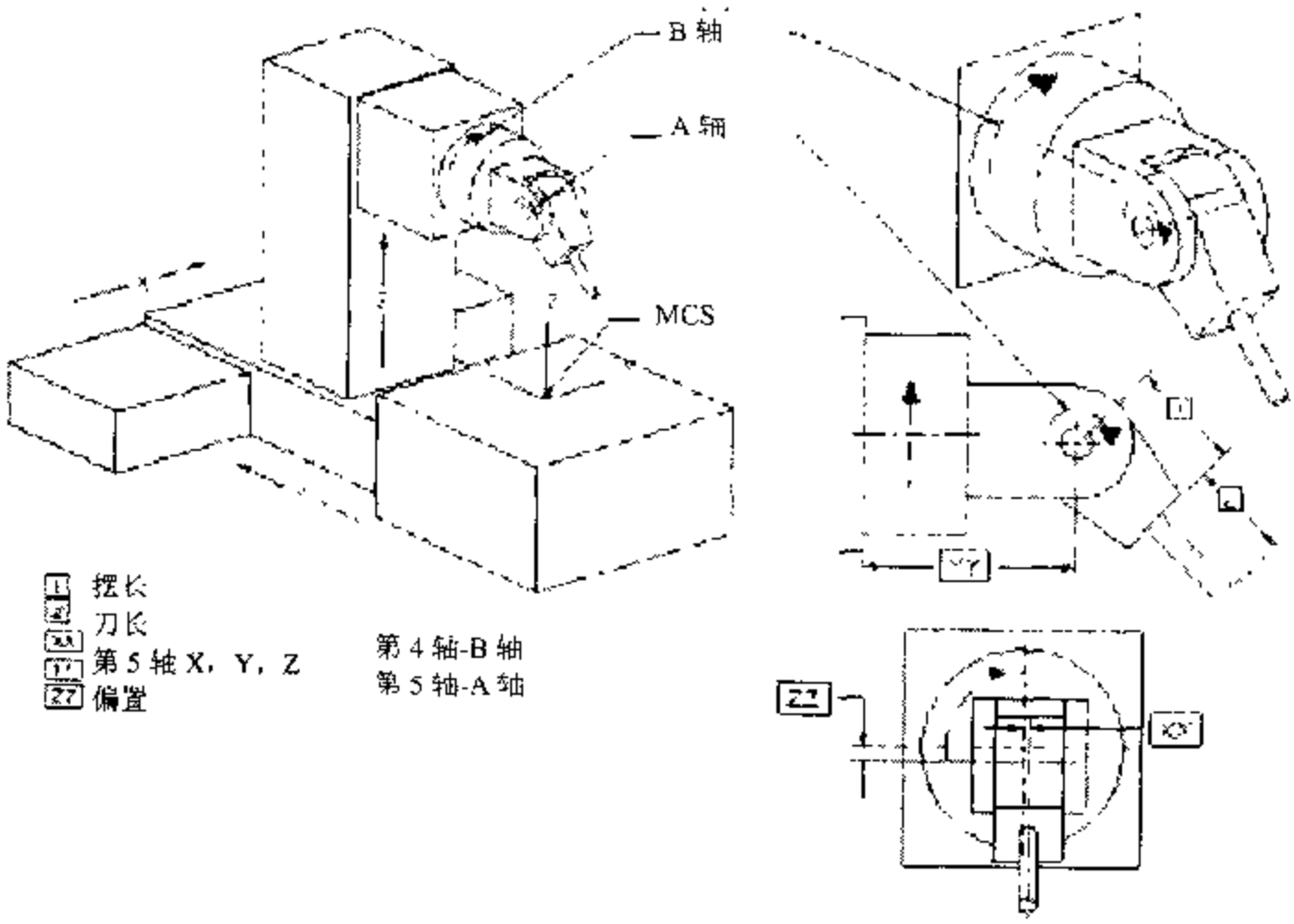


图 9-5 双摆头五轴机床

```
set mom_kin_machine_type                    5_axis_dual_head
set mom_kin_4th_axis_plane                  ZX
set mom_kin_4th_axis_center_offset(0)       0.0
set mom_kin_4th_axis_center_offset(1)       0.0
set mom_kin_4th_axis_center_offset(2)       0.0
set mom_kin_4th_axis_rotation               standard
set mom_kin_4th_axis_zero_position          0.0
set mom_kin_pivot_gauge_offset              100.00
set mom_kin_5th_axis_plane                  YZ
set mom_kin_5th_axis_center_offset(0)       XX(5th axis x-offest)
set mom_kin_5th_axis_center_offset(1)       YY(5th axis y-offest)
set mom_kin_5th_axis_center_offset(2)       ZZ(5th axis z-offest)
set mom_kin_5th_axis_rotation               standard
set mom_kin_5th_axis_zero_position          0.0
```

图 9-6 双摆头五轴机床运动参数

在图 9-7 中，是一个摆头+转台的 5 轴机床。摆头的零点应该设成-90°。当摆头在 0° 时，刀轴与 Y 轴一致，指向工作台。为了与机床坐标系的 Z 轴一致，变量要设为：

- mom_kin_4th_axis_zero_position　　-90.0
- mom_kin_5th_axis_zero_position　　0.0

如果刀轴方向是(0,0,1)，输出 A 轴的角度应该是 90°，因为零点是-90°（如图 9-7 所示）。

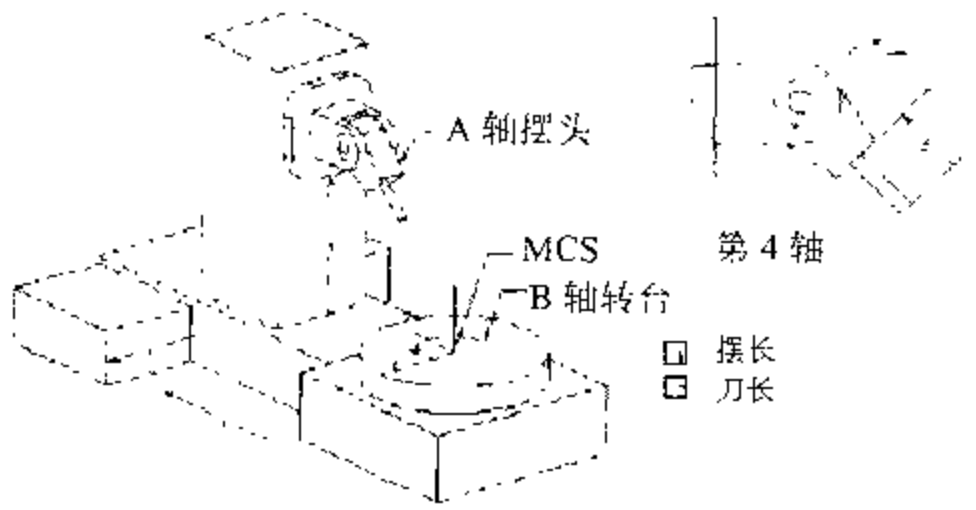


图 9-7　摆头-转台五轴机床

图 9-8 是这个机床的运动参数定义。

```
set mom_kin_machine_type                    5_axis_head_table
set mom_kin_4th_axis_plane                  YZ
set mom_kin_4th_axis_center_offset(0)       0.0
set mom_kin_4th_axis_center_offset(1)       0.0
set mom_kin_4th_axis_center_offset(2)       0.0
set mom_kin_4th_axis_rotation               standard
set mom_kin_4th_axis_zero_position          0.0
set mom_kin_pivot_gauge_offset              100.00
set mom_kin_5th_axis_plane                  ZX
set mom_kin_5th_axis_center_offset(0)       XX
set mom_kin_5th_axis_center_offset(1)       YY
set mom_kin_5th_axis_center_offset(2)       ZZ
set mom_kin_5th_axis_rotation               standard
set mom_kin_5th_axis_zero_position          0.0
```

图 9-8　摆头-转台五轴机床运动参数

2. 摆长

pivot point 是摆轴旋转中心，对于双摆头机床，第 4 轴和第 5 轴可以看作是同一个点。

对于标准的摆头+转台机床和双摆头机床，这个点用下面的参数定义：

- mom_kin_pivot_gauge_offset　　长度

对于特殊机床（如图 9-9 所示），可以定义成 3 个坐标方向的分量：

- mom_kin_gauge_to_pivot(0) XX（X 方向长度）
- mom_kin_gauge_to_pivot(1) YY（Y 方向长度）
- mom_kin_gauge_to_pivot(2) ZZ（Z 方向长度）

在这些机床上，pivot point 在第 4 轴的旋转轴上（如图 9-9 所示）。

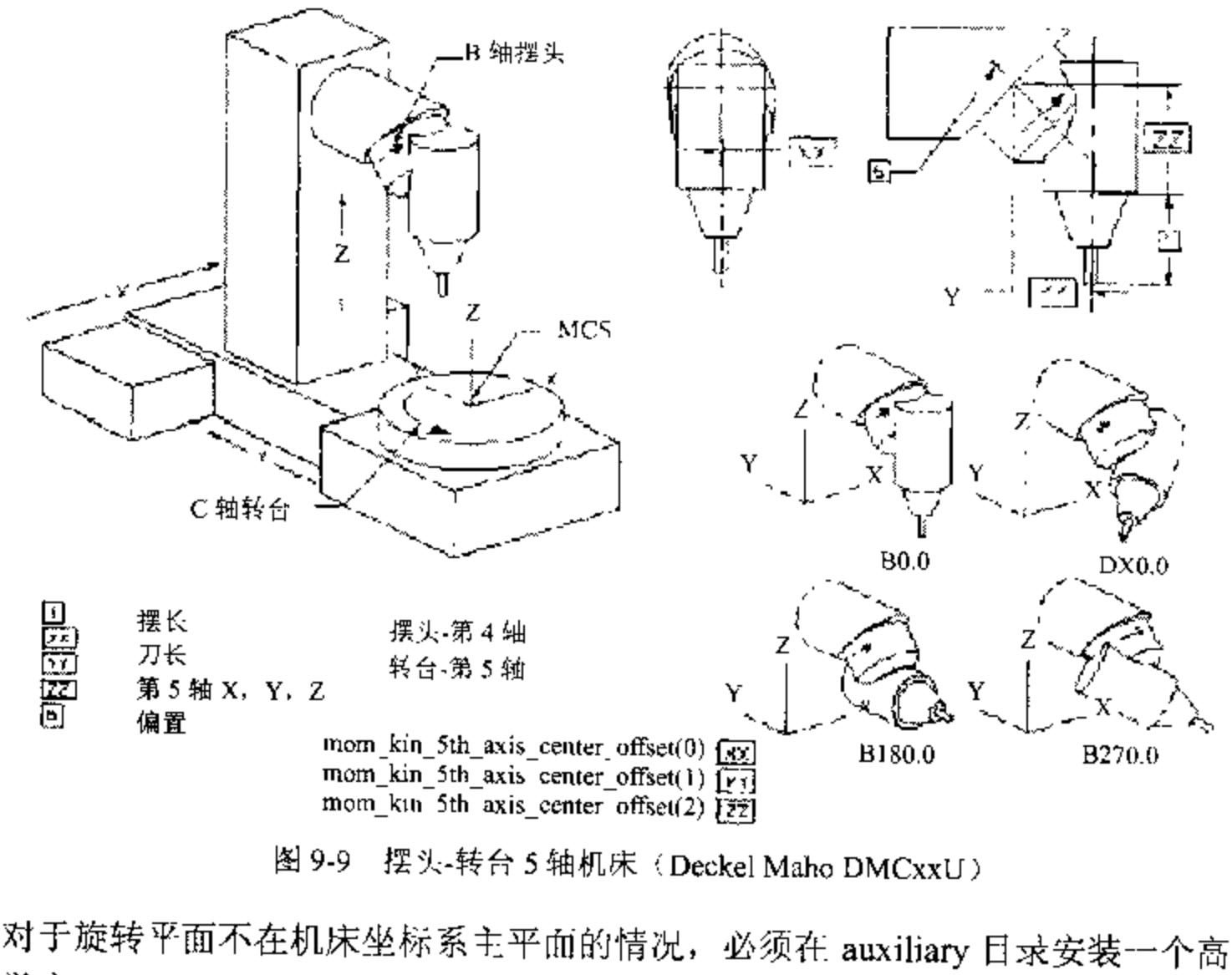


图 9-9 摆头-转台 5 轴机床（Deckel Maho DMCxxU）

对于旋转平面不在机床坐标系主平面的情况，必须在 auxiliary 目录安装一个高级机床运动学库（advanced kinematics library），并在 Tcl 子程序中调用这个库。这个库是个模块选项。

把旋转平面设定为 NONE，而用方向或角度来定义旋转轴的方向（如图 9-10 所示）。

```
set mom_kin_machine_type                5_axis_head_table
set mom_kin_4th_axis_plane              NONE
set mom_kin_4th_axis_center_offset(0)   0.0
set mom_kin_4th_axis_center_offset(1)   0.0
set mom_kin_4th_axis_center_offset(2)   0.0
set mom_kin_4th_axis_rotation           standard
set mom_kin_4th_axis_zero_position      0.0
```

图 9-10 用方向或角度定义

用方向余弦定义（如图 9-11 所示）：

```
set mom_kin_4th_axis_vector(0)          0.0000
set mom_kin_4th_axis_vector(1)          1.0000
set mom_kin_4th_axis_vector(2)          1.0000
```

图 9-11　用方向余弦定义

用两个角度定义（如图 9-12 所示）：

```
set mom_kin_4th_axis_angles(0)          90.0
set mom_kin_4th_axis_angles(1)          45.0
```

图 9-12　用两个角度定义

主轴上刀具安装点（gauge point）到摆头旋转点（pivot point）用变量数组 mom_kin_gauge_to_pivot 定义（如图 9-13 所示）：

```
set mom_kin_gauge_to_pivot(0) XX        (Head Offset X direction)
set mom_kin_gauge_to_pivot(1) YY        (Head Offset Y direction)
set mom_kin_gauge_to_pivot(2) ZZ        (Head Offset Z direction)
set mom_kin_5th_axis_plane              NONE
set mom_kin_5th_axis_center_offset(0)   0
set mom_kin_5th_axis_center_offset(1)   0.0
set mom_kin_5th_axis_center_offset(2)   0.0
set mom_kin_5th_axis_rotation           standard
set mom_kin_5th_axis_zero_position      0.0
```

图 9-13　用变量数组定义

用方向余弦定义（如图 9-14 所示）：

```
set mom_kin_5th_axis_vector(0)          0.0000
set mom_kin_5th_axis_vector(1)          0.0000
set mom_kin_5th_axis_vector(2)          1.0000
```

图 9-14　用方向余弦定义

用两个角度定义（如图 9-15 所示）：

```
set mom_kin_5th_axis_angles(0)          0.0
set mom_kin_5th_axis_angles(1)          0.0
```

图 9-15　用两个角度定义

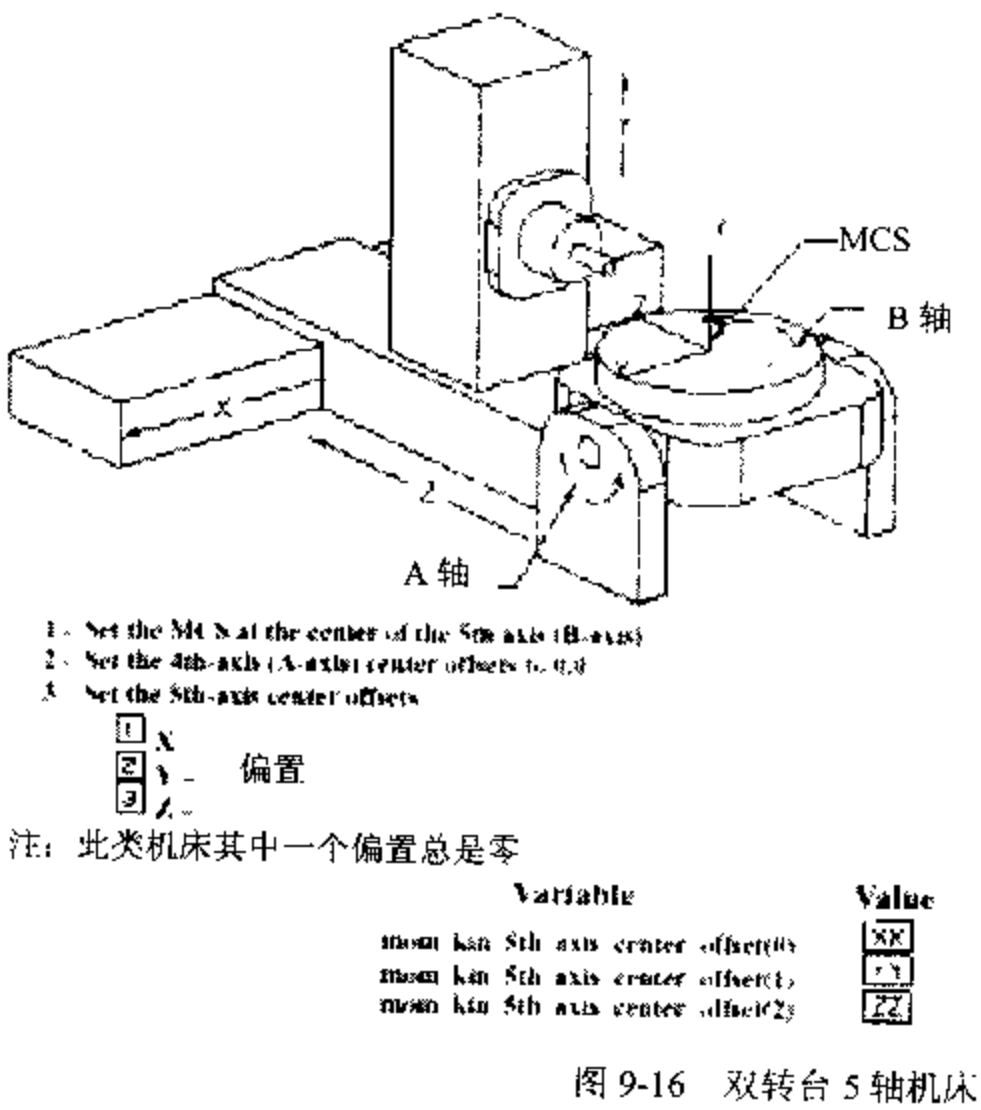
9.4 双工作台机床运动学

9.4.1 普通双转台机床

在图 9-16 中，B 选台装在 A 转台上。建立加工操作时最好把 MCS 定在 B 轴，也就是第 5 轴的中心。

如果机床零点不在 B 轴（第 5 轴）中心，必须把偏差设在 mom_kin_4th_axis _center _offset 中。

第 4 轴旋转中心到第 5 轴旋转中心的偏差设在 mom_kin_5th_axis_center_offset 中（如图 9-16 所示）。



注：此类机床其中一个偏置总是零

图 9-16 双转台 5 轴机床

图 9-17 是这个双转台 5 轴机床的机床运动参数设置。

```
set mom_kin_machine_type              5_axis_dual_table
set mom_kin_4th_axis_plane                YZ
set mom_kin_4th_axis_center_offset(0)     0.0
set mom_kin_4th_axis_center_offset(1)     0.0
set mom_kin_4th_axis_center_offset(2)     0.0
set mom_kin_4th_axis_rotation             standard
set mom_kin_4th_axis_zero_position        0.0
set mom_kin_5th_axis_plane                ZX
set mom_kin_5th_axis_center_offset(0)     XX
set mom_kin_5th_axis_center_offset(1)     YY
set mom_kin_5th_axis_center_offset(2)     ZZ
set mom_kin_5th_axis_rotation             standard
set mom_kin_5th_axis_zero_position        0.0
```

图 9-17　双转台 5 轴机床参数设置

9.4.2　特殊双转台-5 轴机床

对于旋转平面不在机床坐标系主平面的情况，必须把旋转平面设定为 NONE，而用方向或角度来定义旋转轴的方向（如图 9-18 所示）。

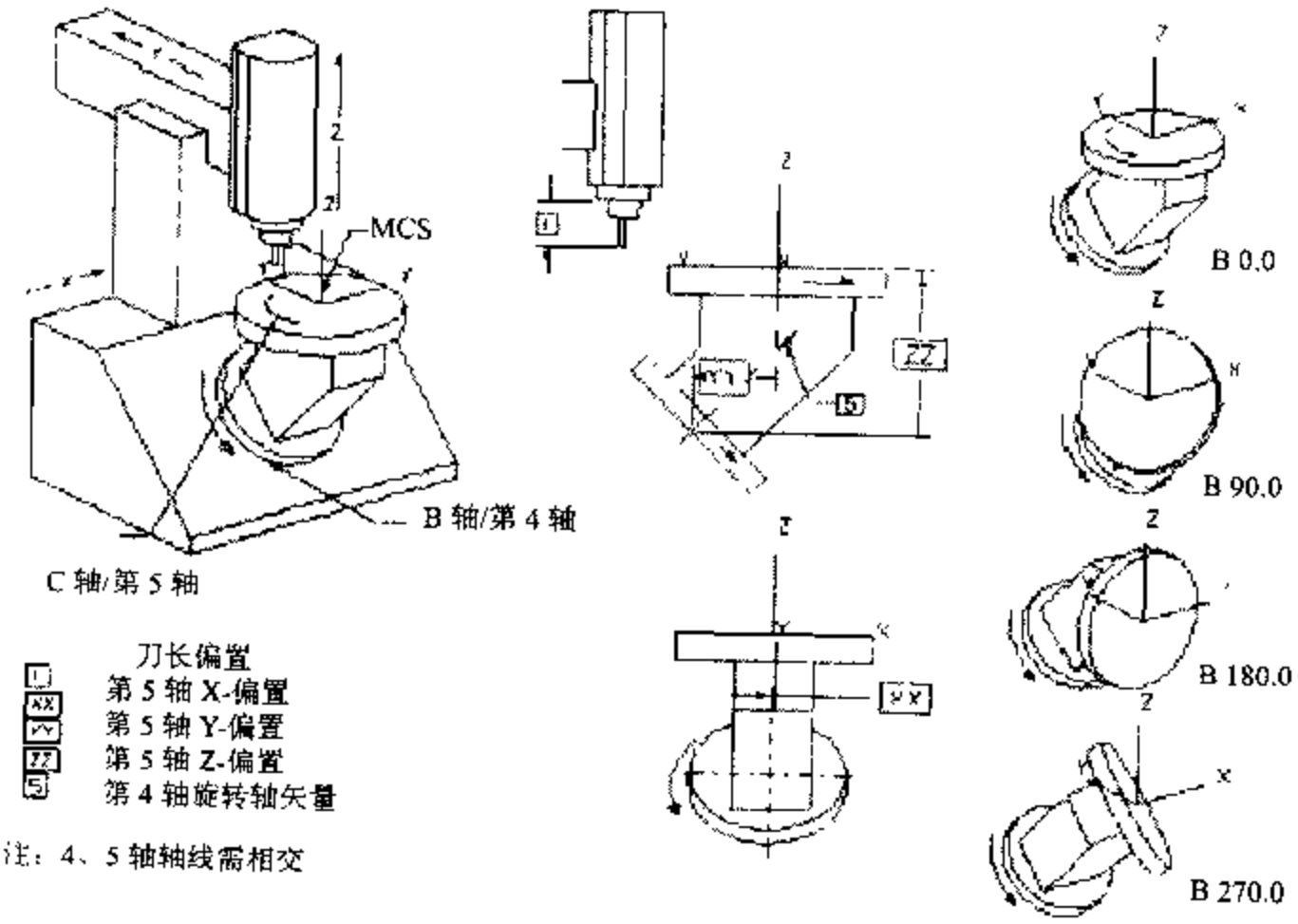


图 9-18　特殊双转台-5 轴机床(Deckel Maho DMUxxV)

图 9-19 是机床类型和第 4 轴的定义。

```
set mom_kin_machine_type                    5_axis_dual_table
set mom_kin_4th_axis_plane                  NONE
set mom_kin_4th_axis_center_offset(0)       0.0
set mom_kin_4th_axis_center_offset(1)       0.0
set mom_kin_4th_axis_center_offset(2)       0.0
set mom_kin_4th_axis_rotation               standard
set mom_kin_4th_axis_zero_position          0.0
```

图 9-19 机床类型和第 4 轴的定义

用方向余弦定义（如图 9-20 所示）：

```
set mom_kin_4th_axis_vector(0)    0.0000
set mom_kin_4th_axis_vector(1)    -1.0000
set mom_kin_4th_axis_vector(2)    1.0000
```

图 9-20 用方向余弦定义

用角度定义（如图 9-21 所示）：

```
set mom_kin_4th_axis_angles(0)          270.0
set mom_kin_4th_axis_angles(1)          45.0
```

图 9-21 用两个角度定义

C 轴旋转中心到 B 轴旋转中心的偏差要设定在数组变量 mom_kin_5th_axis_center_offset 中。一般这两个旋转轴线是共面的。

图 9-22 是第 5 轴的定义。

```
set mom_kin_5th_axis_plane                  NONE
set mom_kin_5th_axis_center_offset(0)       1.0
set mom_kin_5th_axis_center_offset(1)       -100.0
set mom_kin_5th_axis_center_offset(2)       -75.0
set mom_kin_5th_axis_rotation               standard
set mom_kin_5th_axis_zero_position          0.0
```

图 9-22 第 5 轴的定义

用方向余弦定义（如图 9-23 所示）：

```
set mom_kin_5th_axis_vector(0)    0.0000
set mom_kin_5th_axis_vector(1)    0.0000
set mom_kin_5th_axis_vector(2)    1.0000
```

图 9-23　用方向余弦定义

用角度定义（如图 9-24 所示）：

```
set mom_kin_5th_axis_angles(0)          0.0
set mom_kin_5th_axis_angles(1)          0.0
```

图 9-24　用两个角度定义

9.5　在 UG 外部运行 UG/Post

Runugpost 是一个在 UG 外部直接执行 UG/Post 的交互菜单界面。
图 9-25 是文件名和执行指令。

系统	文件名	执行指令
Windows NT	ugpost	ugpost
Unix	ugpost	ugpost

图 9-25　文件名和执行指令

也可以在 Windows 的桌面上选取执行（如图 9-26 所示）。

本章小结

灵活方面的 UG/Post Builder 可以满足大多数铣床和车床的应用需求。对于某些特殊机床，Post Builder 不能满足需求时，需要直接修改定义文件和事件处理文件。本章就是讲述这些内容的。

- 修改定义文件；
- 修改事件处理文件；
- 针对特殊机床的处理；
- 在 UG 外部直接后处理。

图 9-26　在 Windows 的桌面上选取执行

第 10 章　一套渐进式的练习

【目的】

作为对第 9 章的直接修改和用户化 UG/Post 后处理器的阐述的补充，本章将提供一套渐进式的练习，从最基本的事件处理文件和定义文件开始，逐步加入各种机床运动事件，机床控制事件，固定循环事件的处理代码，同时还将对一些特殊事件作进一步的阐述，此外还将使用一些特殊的运用方式，希望能使大家对 UG/Post 的灵活性和扩展性有进一步的认识。

【目标】

通过本章的学习，可以：

- 进一步理解修改 UG/POST 后处理程序的一些方法。

本章包括以下练习：

练习 10-1：创建最基本的事件处理文件和定义文件；
练习 10-2：添加直线运动事件（linear_move）的处理代码；
练习 10-3：添加圆弧运动事件（circular_move）的处理代码；
练习 10-4：添加快速运动事件（rapid_move）的处理代码；
练习 10-5：添加 initial_move 和 first_move 事件的处理代码；
练习 10-6：添加 from_move 和 gohome_move 事件的处理代码；
练习 10-7：添加定义程序头尾格式的处理代码；
练习 10-8：修改行号格式的处理代码；
练习 10-9：添加换刀事件的处理代码；
练习 10-10：添加主轴事件的处理代码；
练习 10-11：添加进给速度的处理代码；
练习 10-12：添加机床控制指令的处理代码；
练习 10-13：添加固定循环的处理代码；
练习 10-14：添加输出刀具信息的处理代码；
练习 10-15：添加输出每一操作加工时间信息的处理代码。

练习 10-1：创建最基本的事件处理文件和定义文件

在这个练习里将从两个空白的文本文件开始，创建最基本的事件处理文件和定义文件

_mypost.tcl 和_mypost.def，主要用于建立 UG/POST 的初始化环境，同时便于理解事件处理文件和定义文件的结构，但并不出现任何输出。

第 1 步　创建最基本的事件处理文件***_mypost.tcl。

- 使用操作系统的文本编辑器创建一个新文本文件***_mypost.tcl。
- 输入以下内容：

```
#-------------------------------------------------------------
#  Date              Revision
#  02-jul-2001 Original

#-------------------------------------------------------------
# The following command invokes the debugging mode.
#-------------------------------------------------------------
# source [MOM_ask_env_var UGII_CAM_DEBUG_DIR]mom_debug.tcl
# source [MOM_ask_env_var UGII_CAM_DEBUG_DIR]mom_review.tcl
# MOM_set_debug_mode ON

#-------------------------------------------------------------
#The following command invokes the warning mode and the default
settings.
#-------------------------------------------------------------
 source [MOM_ask_env_var UGII_CAM_POST_DIR]ugpost_base.tcl

#-------------------------------------------------------------
#The following file consists of default values set for G & M codes
#-------------------------------------------------------------

#------M code declaration

#------G code declaration

#------Kinematic Declaration
 set mom_kin_machine_type                    3_axis_mill
 set mom_kin_machine_resolution              0.0001

#-------------------------------------------------------------
# Post Procedures
#-------------------------------------------------------------
proc MOM_start_of_program {} {
 OPEN_files
 LIST_FILE_HEADER
}
```

```
proc MOM_end_of_program {} {
 LIST_FILE_TRAILER
 CLOSE_files
}
```

第 2 步　创建最基本的定义文件***_mypost.def。

- 使用操作系统的文本编辑器创建一个新文本文件***_mypost.def。
- 输入以下内容：

```
MACHINE mill3ax

INCLUDE
{
   $UGII_CAM_USER_DEF_EVENT_DIR/ude.cdl
}

FORMATTING
{

 WORD_SEPARATOR              ""
 END_OF_LINE          ""
 SEQUENCE Seq_no              1 1 1

 FORMAT  Coordinate           "&__5.3_"
 FORMAT  Feed_MMPM            "&__5_00"
 FORMAT  Feed_MMPR            "&__6.2_"
 FORMAT  Feed                 "&__7.1_"
 FORMAT  Register_2           "&_02_00"
 FORMAT  Seqno                "&__5_00"
 FORMAT  zero                 "&_02_00"
 FORMAT  zero_coord           "&_01.0_"

 FORMAT  EventNum             "%-5d"
 FORMAT  AbsCoord             "%9.4f"
 FORMAT  RotCoord             "%8.3f"
 FORMAT  ComFeed              "%7.2f"
 FORMAT  Rev                  "%5d"
 FORMAT  Minutes              "%9.4f"

ADDRESS N {
 FORMAT Seqno
```

```
 FORCE  off
 MAX    9999
 }

ADDRESS motion_g {
 LEADER "G"
 ZERO_FORMAT zero
 FORMAT Register_2
 FORCE  off
 }

ADDRESS X {
 ZERO_FORMAT zero_coord
 FORMAT Coordinate
 MAX    9999.9999
 MIN    -9999.9999
 FORCE  off
 }

ADDRESS Y {
 ZERO_FORMAT zero_coord
 FORMAT Coordinate
 MAX    9999.9999
 MIN    -9999.9999
 FORCE  off
 }

ADDRESS Z {
 ZERO_FORMAT zero_coord
 FORMAT Coordinate
 MAX    9999.9999
 MIN    -9999.9999
 FORCE  off
 }

ADDRESS F {
 FORMAT Feed
 MAX    9999.9
 MIN    0.1
 FORCE  off
 }
```

```
# Addresses for the lpt file output

 ADDRESS ENUM {
  FORMAT EventNum
  FORCE  always
  LEADER ""
  }

 ADDRESS ABSX {
  FORMAT AbsCoord
  FORCE  always
  LEADER ""
  }

 ADDRESS ABSY {
  FORMAT AbsCoord
  FORCE  always
  LEADER ""
  }

 ADDRESS ABSZ {
  FORMAT AbsCoord
  FORCE  always
  LEADER ""
  }

 ADDRESS AXIS4 {
  FORMAT RotCoord
  FORCE  always
  LEADER ""
  }

 ADDRESS AXIS5 {
  FORMAT RotCoord
  FORCE  always
  LEADER ""
  }

 ADDRESS FEED {
  FORMAT ComFeed
  FORCE  always
 LEADER ""
```

```
 }

ADDRESS RPM {
 FORMAT Rev
 FORCE  always
 LEADER ""
 }

ADDRESS TIME {
 FORMAT Minutes
 FORCE  always
 LEADER ""
 }

BLOCK_TEMPLATE Seq_no  {
 N[$mom_seqnum]
 }

BLOCK_TEMPLATE comment_data {
 ENUM[$mom_debug_event_num]\nows
 " "\nows
 ABSX[$mom_pos (0) ]\nows
 " "\nows
 ABSY[$mom_pos (1) ]\nows
 " "\nows
 ABSZ[$mom_pos (2) ]\nows
 " "\nows
 AXIS4[$mom_pos (3) ]\nows
 " "\nows
 AXIS5[$mom_pos (4) ]\nows
 " "\nows
 FEED[$com_feed_rate]\nows
 " "\nows
 RPM[$mom_spindle_rpm]\nows
 " "\nows
 TIME[$mom_event_time]\nows
 }
}
```

第 3 步　测试结果。

- 在 c:\ugs180\mach\resource\postprocessor\template_post 文件中作如下设置：

```
test,d:\***\***_mypost.tcl,d:\***\***_mypost.def
```

- 在 UG 中打开 pbt_post1.prt 文件。
- 选择操作（operation）p0505，进行后置处理，应无任何内容，也不会出现任何错误提示，结果如图 10-1 所示。

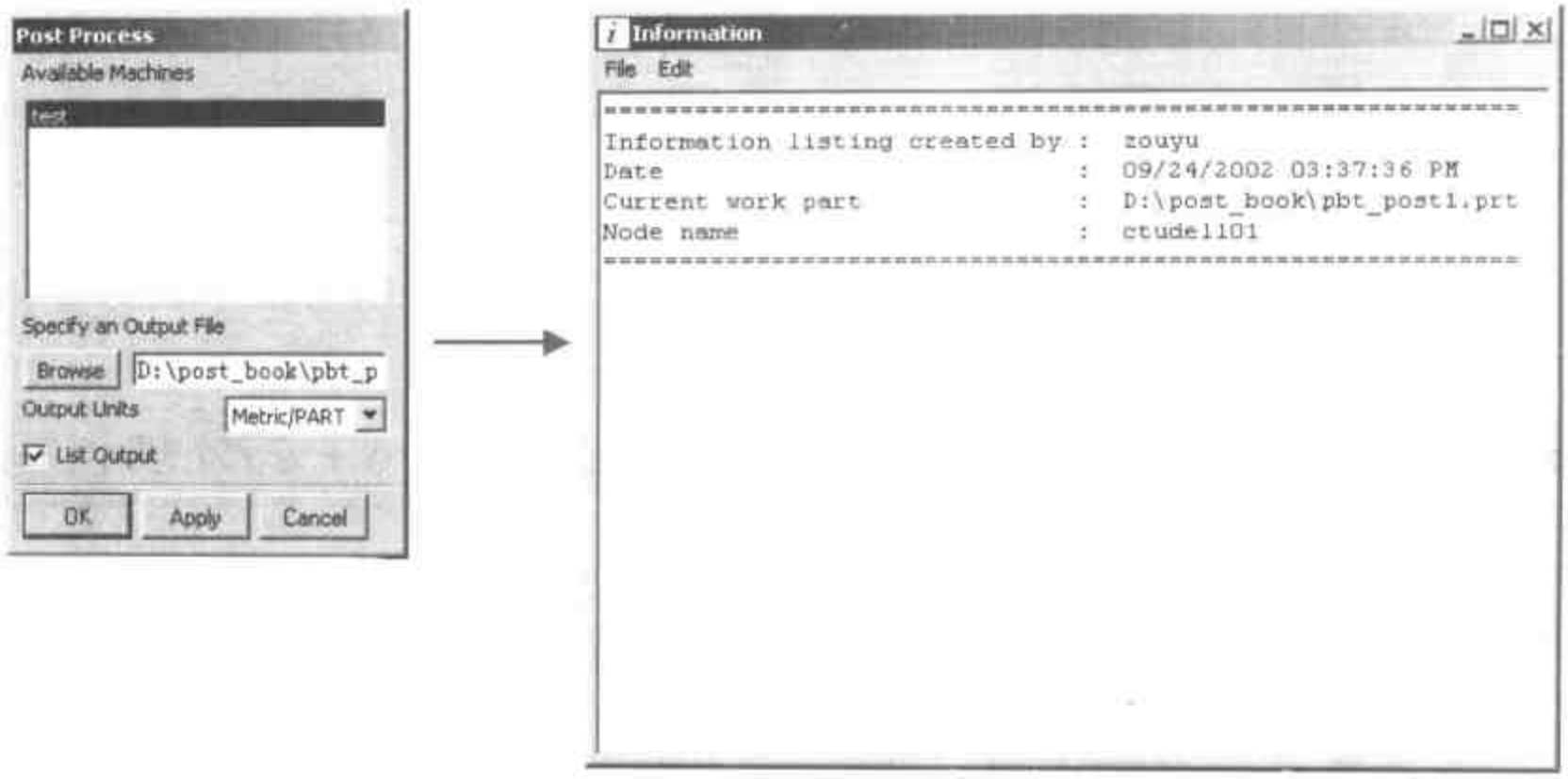


图 10-1　后处理无内容

本练习小结

本练习的结果参见光盘中的 pbt_mypost1.tcl 和 pbt_mypost1.def 文件。

在 TCL 文件中，主要有两大部分内容，一是各种变量的声明（Declaration），主要包括 G 代码声明（G Code Declaration），M 代码声明（M Code Declaration），以及机床运动声明（Kinematic Declaration）；一是各种事件的相应的子程序（proc）。

在本练习的 TCL 文件中，就只有 2 个机床运动方式声明：mom_kin_machine_type 和 mom_kin_machine_resolution，分别定义了机床的类型和机床的精度等级。同时还有最基本的 2 个子程序：MOM_start_of_program 和 MOM_end_of_program。

在 DEF 文件中，主要有 3 大部分内容：BLOCK_TEMPLATE、ADDRESS、FORMAT。BLOCK_TEMPLATE 用于定义输出哪些字地址，ADDRESS 用于定义输出的各个字地址调用的格式，FORMAT 定义了各种格式供 ADDRESS 使用。

在本练习的 DEF 文件中，定义了 2 个 BLOCK_TEMPLATE，以及若干 ADDRESS 和 FORMAT。

因此，整个 TCL 文件和 DEF 文件的总体结构可以理解为图 10-2 所示。

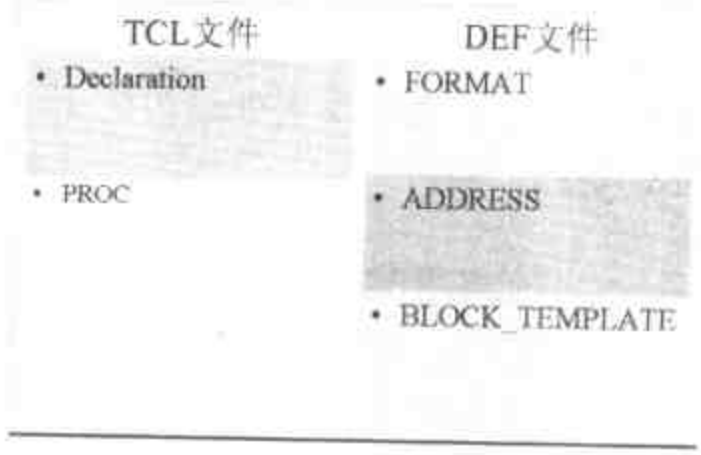


图 10-2　总体结构

练习结束。

练习 10-2：添加直线运动事件（linear_move）的处理代码

在这个练习里，将在前面练习的基础上添加一些代码，用于处理直线运动事件。

第 1 步　在 TCL 文件中，加入 proc MOM_linear_move，如下：

```
proc  MOM_linear_move {} {
 MOM_do_template linear
}
```

第 2 步　在 DEF 文件中，加入 BLOCK_TEMPLATE linear，如下：

```
BLOCK_TEMPLATE linear {
  motion_g[$mom_sys_linear_code]
  X[$mom_pos (0) ]
  Y[$mom_pos (1) ]
  Z[$mom_pos (2) ]
  }
```

第 3 步　在 DEF 文件中，检查相应的 ADDRESS：motion_g，X，Y，Z 是否已正确定义，如下：

```
ADDRESS motion_g {
  LEADER "G"
  ZERO_FORMAT zero
  FORMAT Register_2
  FORCE  off
  }

ADDRESS X {
```

```
 ZERO_FORMAT zero_coord
 FORMAT Coordinate
 MAX    9999.9999
 MIN    -9999.9999
 FORCE  off
 )

ADDRESS Y (
 ZERO_FORMAT zero_coord
 FORMAT Coordinate
 MAX    9999.9999
 MIN    -9999.9999
 FORCE  off
 )

ADDRESS Z (
 ZERO_FORMAT zero_coord
 FORMAT Coordinate
 MAX    9999.9999
 MIN    -9999.9999
 FORCE  off
 )
```

第 4 步　在 DEF 文件中，检查相应的 FORMAT：Zero，Register_2，Zero_coord，Coordinate 是否已正确定义，如下：

```
FORMAT  zero "&_02_00"
FORMAT  Register_2 "&_02_00"
FORMAT  zero_coord "&_01.0_"
FORMAT  Coordinate "&__5.3_"
```

第 5 步　在 TCL 文件中，加入相应的 Declaration，如下：

```
set mom_sys_linear_code          1
```

第 6 步　测试结果。

- 在 UG 中打开 pbt_post1.prt 文件。
- 选择操作（operation）p0505，进行后置处理，结果如图 10-3 所示。

本练习小结

本练习的结果参见光盘中的文件 pbt_mypost2.tcl 和 pbt_ mypost2.def。