



# Agito CNC (插补运动) 使用手册 v1.0



[www.agito-akribis.com](http://www.agito-akribis.com)

Member of Akribis Systems group



## 版本记录

版本	描述	日期
1.0	首版发布	2023/9/19

※本公司保留不定期更新的权利，根据产品硬件及软件的升级或更新迭代以及市场需求，本手册将会不定期进行内容上的更新调整，恕不另行告知，如需最新本本文档，请联系 Agito-Akribis 公司获取相应支持。

## 目录

1	介绍	4
1.1	关于手册	4
1.2	CNC（插补运动）简介	4
1.3	CNC 功能注意事项	4
2	操作步骤	5
2.1	PCSuite CNC 界面介绍	5
2.2	CNC Segment 说明	7
2.3	如何使用 CNC	15
2.3.1	示例 1——在 PCSuite CNC 界面操作	15
2.3.2	示例 2——在 IDE UserProgram 环境使用	17
2.3.3	示例 3——导入 G/M-Code	22

# 1 介绍

---

## 1.1 关于手册

---

感谢您选择 Agito 系列运动控制产品，我们将竭力为您提供追求速度与精度的极致运动控制方案，并提供全方位的技术支持。

本手册主要介绍 Agito CNC (插补运动) 功能使用方法。

手册中仅详细介绍与 CNC 运动相关的配置内容，其他参数设置可参阅《Agito 快速入门手册》及其他相关手册中的详细介绍，本文档将不再赘述。

## 1.2 CNC（插补运动）简介

---

CNC (Computerized Numerical Control) 是控制器按事先存储的程序来执行预定义的轨迹和外设的操作逻辑控制功能，通过控制器算法生成的微观指令传送给伺服驱动电机或液压等执行元件代送设备运行。

最大支持 8 轴协调动作。

支持 G-Code 及 M-Code 导入。

## 1.3 CNC 功能注意事项

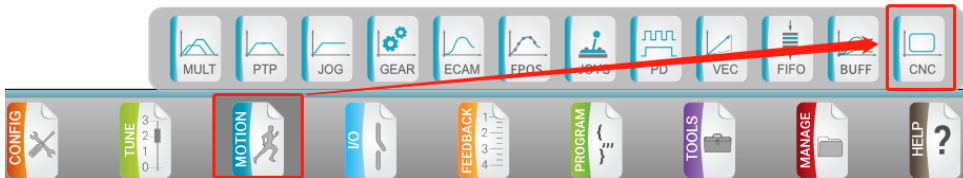
---

- 不同控制器轴数有差异；
- 在使用 CNC 功能前请确保各轴参数已正确设置并且电机可以正常运动；
- CNC 中的速度都是指的矢量速度，即多轴的合速度

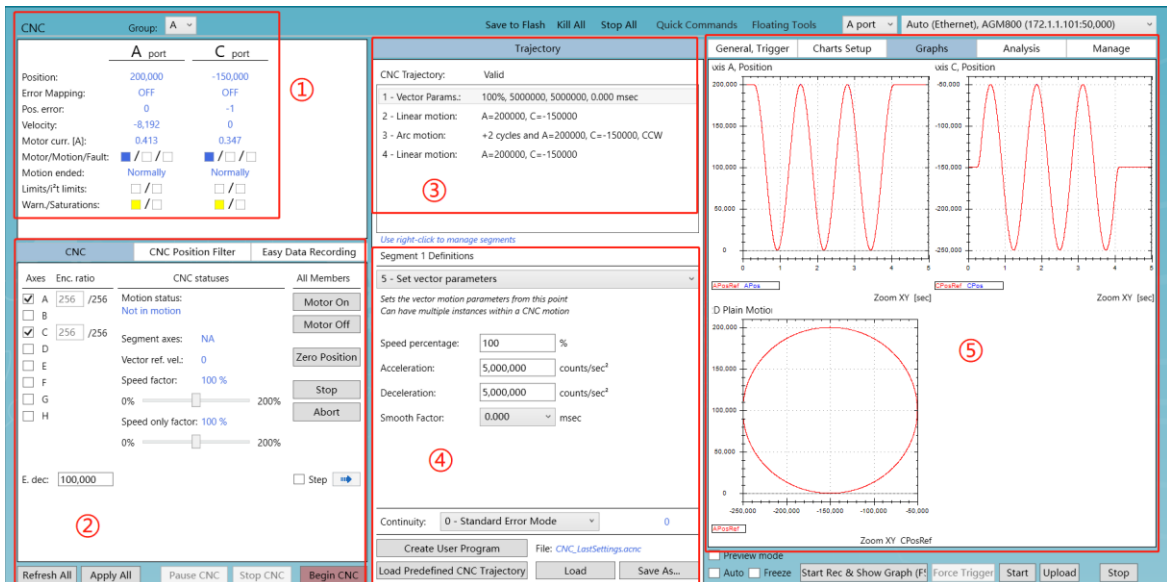
## 2 操作步骤

Agito PCSuite 提供一个方便 CNC 程序编辑界面，用户可以在该界面编写所需的 CNC 程序段，并且可以生成 IDE 程序在 IDE+编程环境中运行，实现用户各种运动轨迹。

### 2.1 PCSuite CNC 界面介绍

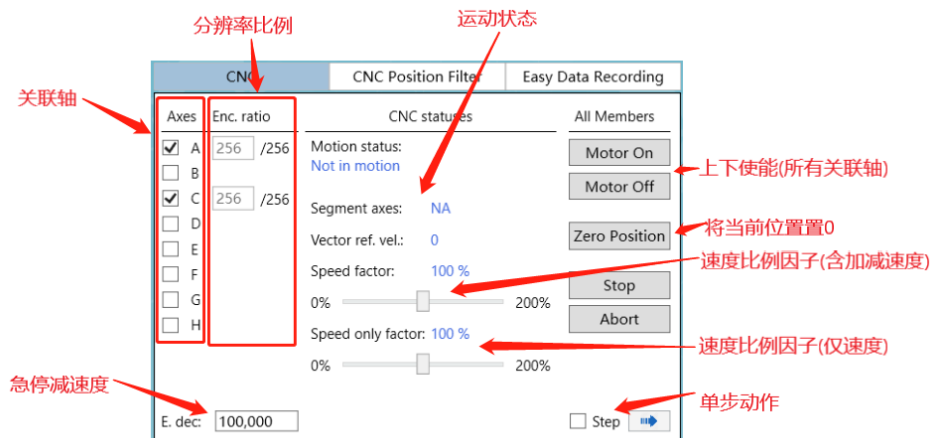



1. 切换菜单栏到 MOTION ---> CNC 界面如下：



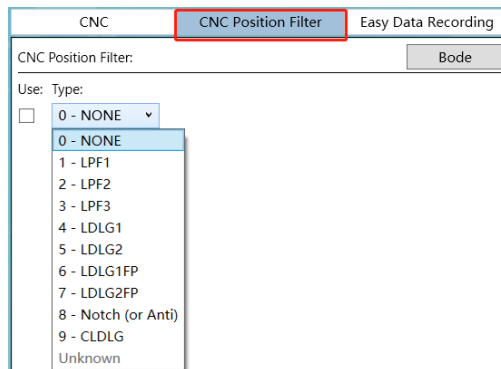
- ◆ ①关联轴状态窗：将根据②中左侧勾选的关联轴显示当前 CNC 关联轴运动状态，  
Group: CNC 组，可分 2 组，A 和 B，例如 ACNC $\underline{A}$ EncRatio 中的“ $\underline{A}$ ”即表示 Group A；
- ◆ ②CNC 设置窗：此处用于设置该组 CNC 轴各运动之间的关联参数，以及滤波、抓图等参数设置，有 3 个页面，

#### 页面 1: CNC 参数

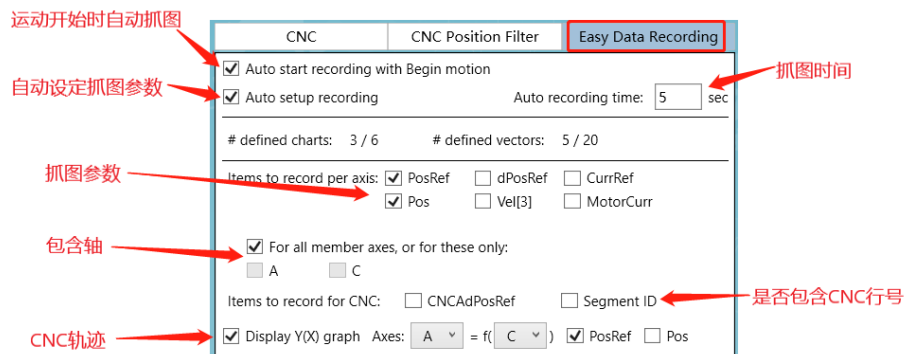


- **Axes:** 选择该组 CNC 关联轴，不同控制器所支持轴数有差异，如图示中的 AGM800 最大支持 8 轴 CNC；
- **Enc.ratio:** [CNC $\Delta$ EncRatio] ( $\Delta$ 表示 Group 号，下同)，表示相对分辨率比例，注意：**此功能需要使用定制固件支持**；
- **Speed Factor:** [CNCAPercents]，CNC 矢量速度因子，加减速度均受该因子影响，以确保运动时间与该因子成正比；
- **Speed only factor:** [CNCASpeedPer]，CNC 矢量速度因子，但加减速度不受该因子影响；
- **Zero Position:** 将当前 CNC 所有关联轴位置置 0（即 SetPosition=0）；
- Step ：单步动作，勾选将会逐行执行 CNC 指令；

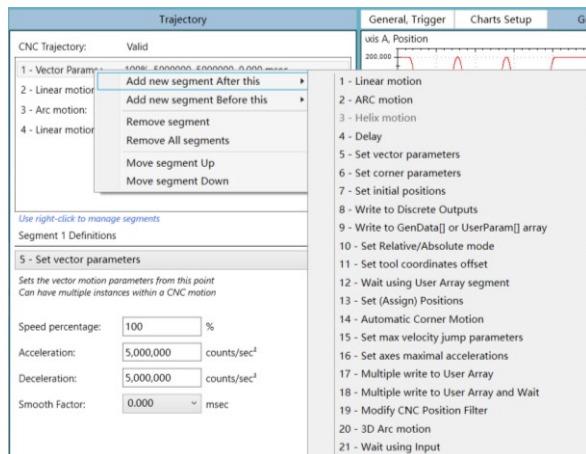
## 页面 2：滤波器及伯德图



## 页面 3：示波器抓图



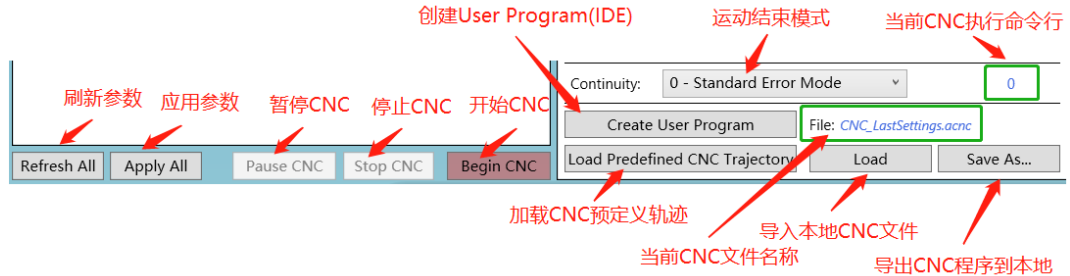
- ◆ ③CNC 运动轨迹步骤：用户在此界面编辑 CNC 运动轨迹命令行；



- 鼠标右击 CNC 指令行，可以增加或删除指令行；
- 指令有 21 种可供选择（3-Helex motion 在开发中，暂不支持使用）；
- 鼠标左键单击已加入的 CNC 命令行，在下面参数设定区域可以进行修改命令行及其对应参数等操作；

- ◆ ④运动轨迹参数：此处显示所选择的 CNC 命令行的参数设置区，下节将会对每种指令进行相应的解释，请参阅相关内容；

- ⑤运动轨迹抓图：如选择自动抓图模式，此处将根据用户设定参数抓取相应的参数图像（关于抓图设置请参阅上页②-页面 3 内容（Page 6））所述；
- CNC 操作菜单栏

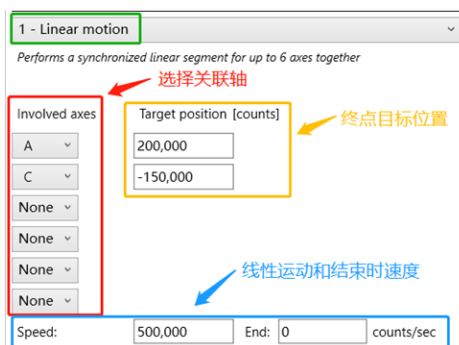


## 2.2 CNC Segment 说明

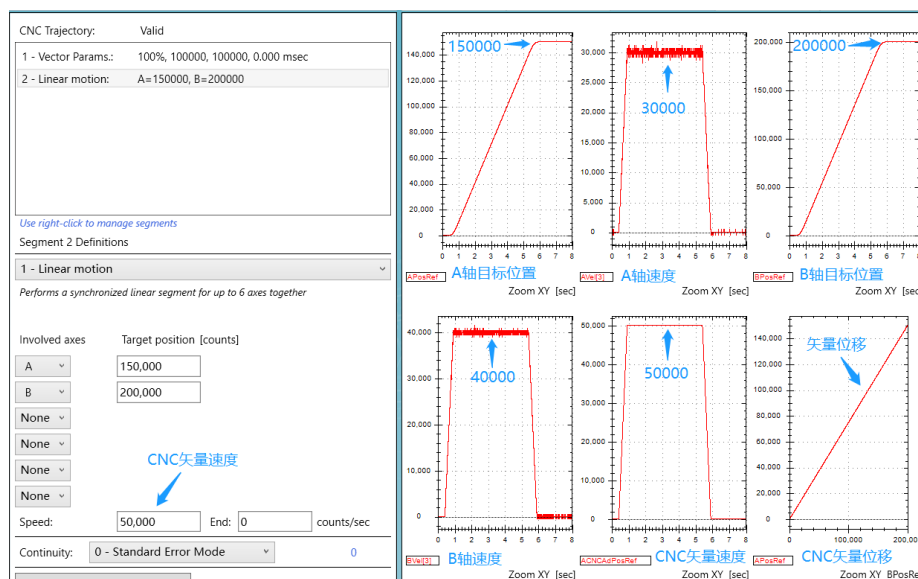
Agito CNC 运动中共有 21 种命令可供选择（由于 PCSuite 版本和控制器型号及固件版本差异，指令类型数量可能存在差异，部分命令开发中，暂不支持使用），以下进行详细描述：

### 1 - Linear motion

**线性运动：**选择运动轴（不同型号控制器所支持轴数有差异），各轴的终点位置，以及运动过程中及结束时的速度；



例如以下示例中：(APos,BPos)分别从(0,0)位置运动到(150000,200000)的位置处， $CNCspeed = \sqrt{Avel^2 + Bvel^2} = \sqrt{30000^2 + 40000^2} = 50000$ ;

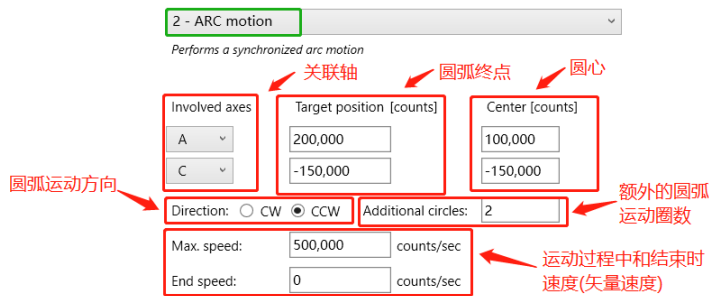


## ◆ 2 - ARC-motion

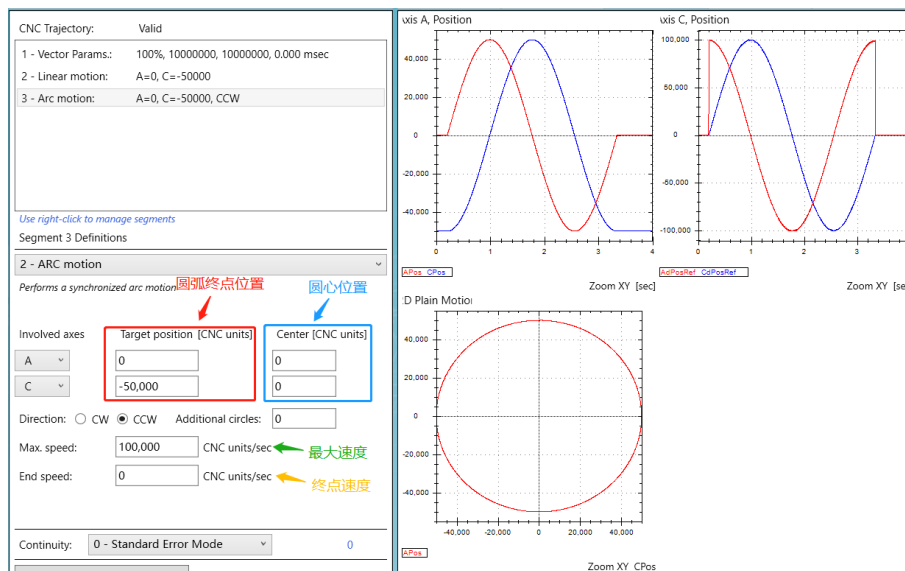
**圆弧运动：**选择圆弧运动关联轴（2 轴），圆弧轴终点及圆心位置，运动方向，圈数（注意：所填入值为额外的圆弧圈数，如值为 2 则实际运动 3 圈）

整圆起点和终点位置相同，非整圆的圆弧的起点则为上一行 CNC 运动终点，终点由圆弧运动中的终点定义，特别注意圆心位置；

注意：目前圆弧运动不能作为 CNC 运动第 1 段，必须先运动到圆弧的起点位置；



例如以下示例中：弧线(APos,BPos)从起始点(0,-50000)的位置以圆心(0,0)运动到(0,-50000)位置处，即半径 R 为 50000 的整圆， $APos=R*\sin(t)$ ， $BPos=R*\cos(t)$ ；

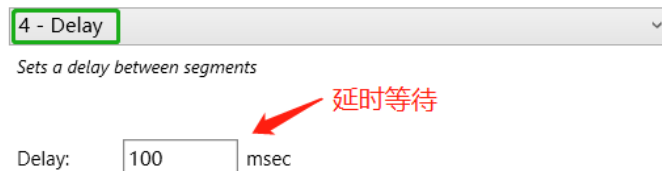


## ◆ 3-Helix motion

**螺旋线：**开发中，暂不支持使用；

## ◆ 4 - Delay

延时等待，上一行命令和下一行命令之间延时时间，其前一运动段结束速度必须为 0；



## ◆ 5 - Set vector parameters

**矢量参数：**设定运动矢量参数；

影响其后所有运动，直到扫描到另一个“Set vector parameters”程序段为止；



5 - Set vector parameters

*Sets the vector motion parameters from this point  
Can have multiple instances within a CNC motion*

Speed percentage:  %

Acceleration:  counts/sec<sup>2</sup>

Deceleration:  counts/sec<sup>2</sup>

Smooth Factor:  msec

速度比例

加速度

减速度

平滑时间

◆ 6 - Set corner parameters

倒角参数：用于设定该行之后“14 - Automatic Corner Motion”的倒角参数，倒角包含两种类型：ARC(圆弧)和 Continuous accel(连续加速)

➤ ARC(圆弧)类型

6 - Set corner parameters

*Defines the general corner parameters (type, radius...) to use in all following Auto Corner Motion segments*

Corner type:

Corner radius method:  Radius  Error

Corner radius:  counts

Corner minimal angle:  deg.

Axis accel. limit type:  Avoid  Use

倒角类型

倒角方式

倒角半径

最小角度

轴加速度限制

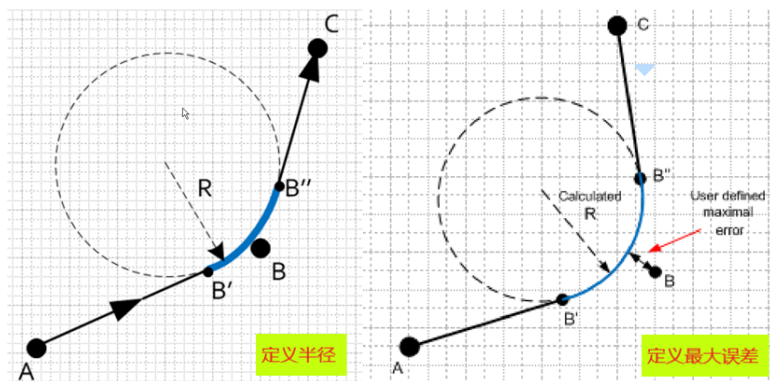
对于圆弧类型，有 2 种倒角方式：定义半径和定义最大误差

定义半径方式下：

Corner radius：表示圆弧倒角半径 R，

Corner minimal angle：表示开启自动圆弧倒角的最小线夹角，例如下左图中：假设  $\overline{AB}$  和  $\overline{BC}$  之间夹角为  $45^\circ$ ，则填入此处的最小角度必须  $<45^\circ$  自动倒角才能生效。

定义最大误差方式下：



➤ Continuous accel(连续加速)类型

6 - Set corner parameters

*Defines the general corner parameters (type, radius...) to use in all following Auto Corner Motion segments*

Corner type:

Corner axis acceleration:  counts/sec<sup>2</sup>

Corner minimal angle:  deg.

Axis accel. limit type:  Avoid  Use

#### ◆ 7 - Set initial positions

该命令开发中，暂不支持使用；

7 - Set initial positions

*Sets the expected initial positions of all the member axes  
Once per a CNC motion, first segment*

Involved axes	Initial position [counts]
A	<input type="text" value="0"/>
C	<input type="text" value="0"/>
None	
None	
None	
None	

位置设定值

关联轴

#### ◆ 8 - Write to Discrete Outputs

写入数字输出状态：该命令用户操作数字输出口状态（例如图示将 DO\_1、DO\_2、DO\_3 置为“On”状态）

8 - Write to Discrete Outputs

*Writes a specified value to DOutPort of a given port*

Axis:  DOutPort value:

操作数字输出

Floating Window - Discrete Outputs: AGM800 @Auto (... X)

A port Discrete Outputs

1  2  3  4

(check to activate/deactivate)

#### ◆ 9 - Write to GenDatall or UserParam[] array

数组写入命令：该命令用于写入 GenData 或 UserParam 值，可用于在 CNC 程序运行时，触发其他线程中的任务；

9 - Write to GenData[] or UserParam[] array

*Writes a specified value to GenData[] or UserParam[] array at a specified index*

Array to use:  GenData  UserParam

GenData [  ] =

#### ◆ 10 - Set Relative/Absolute mode

该命令开发中，暂不支持使用；

◆ **11 - Set tool coordinates offset**

**设定偏移量：** 该指令用于设定运动过程中，绝对目标位置的偏移量；

11 - Set tool coordinates offset

*Defines the offsets to add to all absolute targets positions (over all the CNC trajectory). Applies only when in Absolute mode*

Involved axes	Offset [counts]
A	5
B	20
None	
None	
None	
None	

◆ **12 - Wait using User Array segment**

**条件等待：** CNC 运行到该行时，判断设定条件是否满足，不满足停在该行等待；

12 - Wait using User Array segment

*When reaching this type of segment, the CNC engine will wait, in-position of all member axes, till the trigger condition as defined by the segment's parameters is satisfied*

Array to use:  GenData  UserParam ↖ 数组

Wait for parameter: GenData [ 1 ]

Trigger to wait for ↖ 触发条件

Trigger type: 2 - Equal To

Trigger value: 100

◆ **13 - Set (Assign) Positions**

**位置设定：** 该命令用于将当前位置设定为用户所需位置值（如图示中将 A、C、D 轴当前位置置 0，即 Pos=0）

13 - Set (Assign) Positions

*Assigns new position values to selected member axes (must not be moving)*

Involved axes	Assign position [counts]
A	0
C	0
D	0
None	
None	
None	

↖ 位置设定值

◆ **14 - Automatic Corner Motion**

**自动倒角：** 该命令用于自动倒角，其参数由“6 - Set corner parameters”决定，值得注意的是，该命令必须放置于 2 条线性运动之间，并且只能选择 2 关联轴；

14 - Automatic Corner Motion

Request for automatic calculation of a corner, based on the parameters defined in the last Set Corner Parameters segment

Involved axes

A

C

关联轴

◆ 15 - Set max velocity jump parameters

**最大速度跳跃设定：**该命令用于定义两个线性运动（Linear motion）之间每个轴最大速度跳变（两线性运动之间没有自动倒角），

如果该程序段识别出两个线性运动段之间的速度差值超过所允许的最大速度差值，该程序段将会改变线性运动的结束速度；

15 - Set max velocity jump parameters

Define the maximal velocity jump per axis, where a Linear Motion segment is followed by another Linear Motion segment (without an Automatic Corner Motion segment between them)

Velocity jump mode:  Ignore  Use

Involved axes      Max. Vel. Jump [counts/sec]

A	5,000
C	5,000
D	10,000
None	
None	
None	

◆ 16 - Set axes maximal accelerations

**设定轴最大加速度：**注意该命令仅用于自动倒角参数中；

16 - Set axes maximal accelerations

Used only if enabled at Corner Parameters segment

Involved axes      Max. Accel. [counts/sec<sup>2</sup>]

A	10,000
C	10,000
D	50,000
None	
None	
None	

◆ 17 - Multiple write to User Array

**多数组值写入：**该命令可同时向 4 个连续数组（GenData 或 UserParam）进行值写入操作；

17 - Multiple write to User Array

Writes 4 values to GenData[] or UserParam[] array at 4 consecutive indexes

Array to use:  GenData  UserParam

Array items to write

GenData	[	<input type="text" value="200"/>	]	=	<input type="text" value="0"/>
	[	<input type="text" value="201"/>	]	=	<input type="text" value="0"/>
	[	<input type="text" value="202"/>	]	=	<input type="text" value="0"/>
	[	<input type="text" value="203"/>	]	=	<input type="text" value="0"/>

◆ 18 - Multiple write to User Array and Wait

条件等待并写入参数：当满足所设定条件时（例如图示 AGenData[100]=1 时）触发 4 个连续数组值写入操作，不满足条件时，停在此行等待；

18 - Multiple write to User Array and Wait

Writes 4 values to GenData[] or UserParam[] array at 4 consecutive indexes, then wait, in-position of all member axes, till the trigger condition as defined by the segment's parameters is satisfied

Array to use:  GenData  UserParam

Array items to write before waiting	Trigger to wait for
[ <input type="text" value="200"/> ] = <input type="text" value="0"/>	Array index: <input type="text" value="100"/>
[ <input type="text" value="201"/> ] = <input type="text" value="0"/>	Trigger type: <input type="text" value="2 - Equal To"/>
[ <input type="text" value="202"/> ] = <input type="text" value="0"/>	Trigger value: <input type="text" value="1"/>
[ <input type="text" value="203"/> ] = <input type="text" value="0"/>	

触发条件

被写入数组

◆ 19 - Modify CNC Position Filter

开启或关闭 CNC 运动过程中的位置滤波器；

19 - Modify CNC Position Filter

Enables and sets/modifies the CNC Position Filter parameters.  
(Must not be moving)

CNC Position Filter:  Enabled  Disabled

Filter type:

- NONE
- LPF1
- LPF2
- LPF3
- LDLG1
- LDLG2
- LDLG1FP
- LDLG2FP
- Notch (or Anti)

Continuity:  or Mode

◆ 20 - 3D Arc motion

3 维圆弧运动：该命令用于执行 3 维同步圆弧运动，用法和 2 维圆弧类似，

值得注意的是，如果圆弧运动起点位置、目标位置和圆心共线（如下图 2 所示，圆弧轨迹为半球面），则需要填入圆弧上附加位置（Additional），

如果圆弧为非半球面圆弧，则将附加位置（Additional）填 0；

20 - 3D Arc motion

Performs a synchronized 3D arc motion.  
If the start point, the target position and the center are colinear, additional point on the arc are required, otherwise the additional column should be set to 0.

Involved axes	Target position	Center	Additional (all in [counts])
A	0	0	100
C	100	0	0
D	0	0	0

Direction:  CW  CCW Additional circles:

Max. speed:  counts/sec ← 最大速度

End speed:  counts/sec ← 结束速度

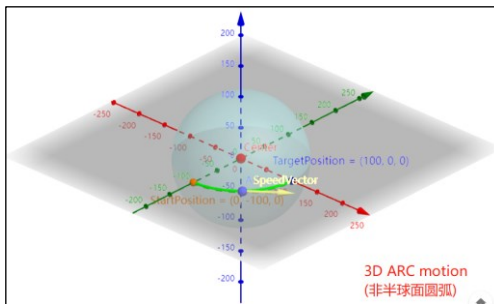


图 1

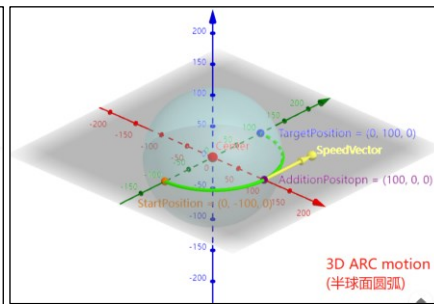


图 2

3D Arc 圆弧需要注意其位置需要满足球体关系式，

例如起点位置为(APos1, BPos1, CPos1)，圆心(Center)为(a, b, c)，目标位置(Target position)为(Apos2, Bpos2, Cpos2)，需要满足以下关系：

$$((APos1 - a)^2 + (BPos1 - b)^2 + (CPos1 - c)^2) = ((Apos2 - a)^2 + (Bpos2 - b)^2 + (Cpos2 - c)^2),$$

其矢量偏差 < 5counts.

#### ◆ 21 - Wait using Input

**条件等待：**当 CNC 扫描到该行时，将在此处等待，直到满足触发条件继续往后执行，支持数字信号和模拟信号，其前一运动段结束速度必须为 0；

例如图示中，等待 Digital\_Input\_3 为 “On” 状态时触发 CNC 继续往后执行；

21 - Wait using Input

When reaching this type of segment, the CNC engine will wait, in-position of all member axes, till the trigger condition as defined by the segment's parameters is satisfied

Input to use:  Discrete  Analog

Wait for parameter: A DInPort & 4

Trigger to wait for

Trigger type: 2 - Equal To

Trigger value: 4

## 2.3 如何使用 CNC

### 2.3.1 示例 1——在 PCSuite CNC 界面操作

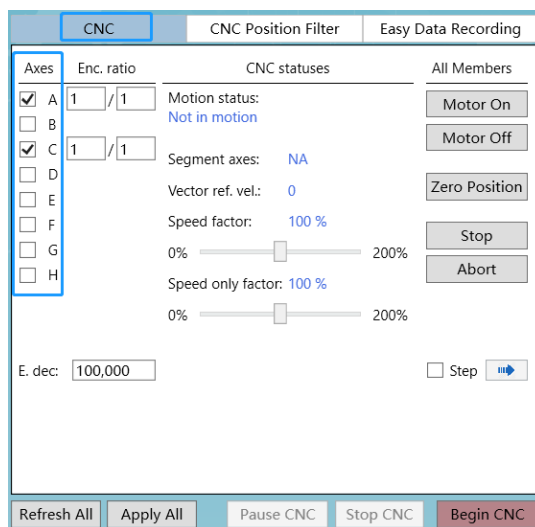
本例以 AGM800+AGA 控制系统作为展示，运动轨迹为一个“操场”的形状：

方式 1)：按顺序自动执行

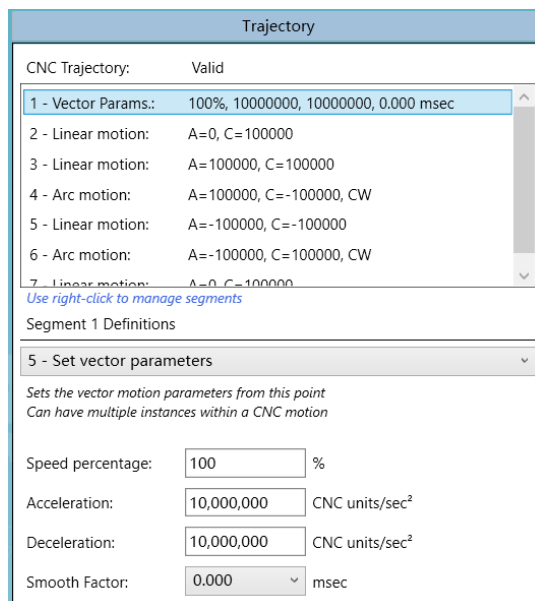
- 第 1 步：切换菜单栏到 **MOTION ---> CNC** 界面



- 第 2 步：在 CNC 设置窗勾选 CNC 关联轴（本例中使用 A、C 两轴）

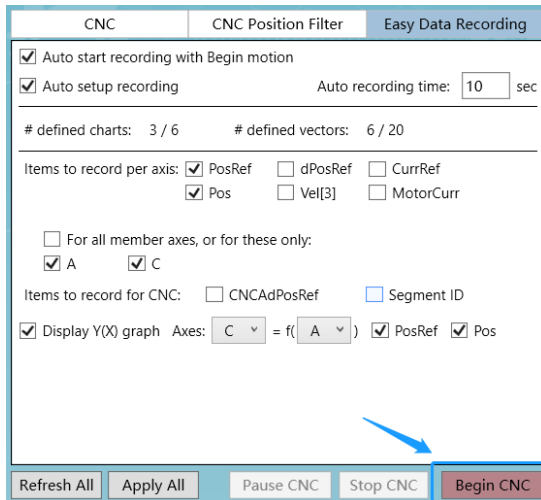


- 第 3 步：在“Trajectory”界面上自而上而下依次增加并编辑 CNC 指令行；




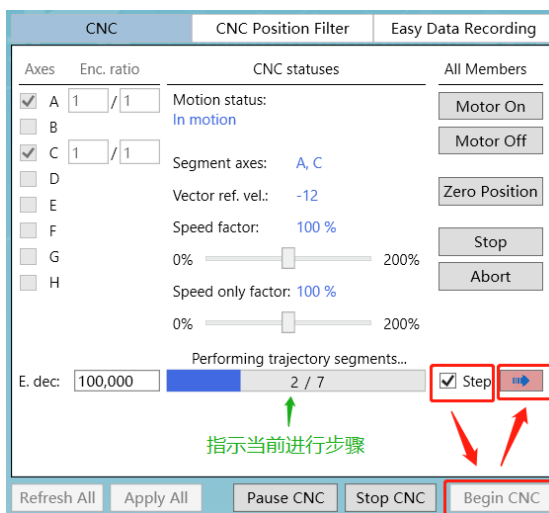
- 第 4 步：点击“Begin CNC”，控制器开始按顺序执行 CNC 指令；

在“Begin CNC”前可以设置好抓图相关参数，可以从图像中运动轨迹（注：Display Y(X)可以绘制出 CNC 2D 轨迹图）

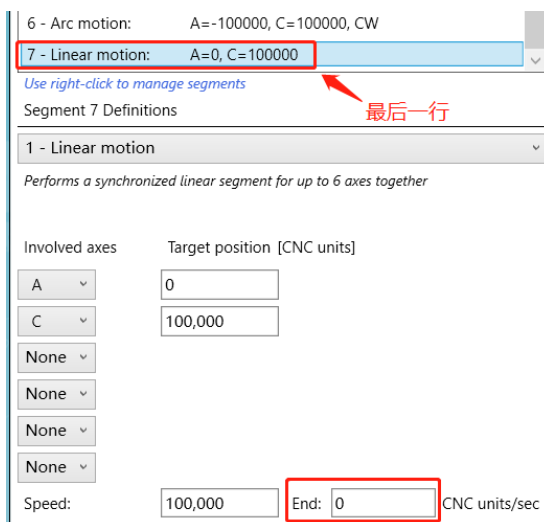


方式 2)：按顺序逐步执行（通常用于调试）

- 步骤 1~3 和方式 1) 相同；
- 步骤 4: 勾选 “ Step ” ----> “Begin CNC” ----> “ ”，可按顺序逐步执行 CNC；



值得注意的是 CNC 最后一行运动指令的结束速度（End speed）必须为 0:



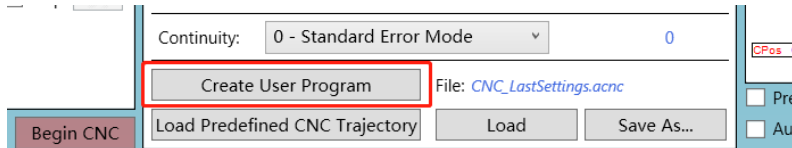


### 2.3.2 示例 2——在 IDE UserProgram 环境使用

当用户需要在 IDE UserProgram 中运行 CNC 程序时，可以先在 CNC 界面编辑并调试好程序，确定程序运行正常无错误后，导出成 UserProgram 嵌入用户的程序中。当然也支持用户在 UserProgram 中直接编写 CNC 程序段，前提是用户对 CNC 程序在 IDE 中的参数命令足够熟悉，否则很容易出现错误导致程序无法正常运行。

本例中以从 CNC 中导出示例 1 中的 CNC 轨迹为例，并对导出的 UserProgram 中的命令参数做解释来为帮助客户深入了解 CNC 在 IDE 环境中的应用。

- 第 1 步：在 CNC 界面中将编辑调试好的 CNC 轨迹导出成 UserProgram；



- 第 2 步：将上一步导出的 CNC UserProgram 嵌入到用户程序中；

以下截取了导出 UserProgram 的部分程序段：

每一小段前面都会有一个简要的注释，用于标注该 CNC 段的含义

```

13 // Encoder ratios
14 //
15 ACNCAEncFactNu = 1
16 ACNCAEncFactDn = 1
17 CCNCAEncFactNu = 1
18 CCNCAEncFactDn = 1
19 //
20 // Vector speed percents
21 //
22 CCNCAPercents = 100
23 //
24 // Vector speed only percents
25 //
26 CCNCASpeedPer = 100
27 //
28 // Clear CNCA FIFO
29 //
30 CCNCAclear
31 //
32 // Vector Motion Continuity Mode
33 //
34 ACNCAEndSegMod=0
35 //
36 // Segment index: 1
37 //
38 // Push Vector Params.
39 // Percents: 100%
40 // Acceleration: 10000000
41 // Deceleration: 10000000
42 // Smooth Factor: 0
43 //
44 CCNCAPushType,100663295
45 CCNCAPushParam,100
46 CCNCAPushParam,10000000
47 CCNCAPushParam,10000000
48 CCNCAPushParam,0
49 //
50 // Segment index: 2
51 //

```

新的 CNC 段可以直接衔接在上一段 CNC 段之后，在插入 Segment 之前注意关联轴 MotionMode 设为

- 以下关键字用于定义 CNC FIFO 参数:

- ❖ **CNCAPushType**: 用于在 CNC FIFO 中初始化一个新“段” (segment)。

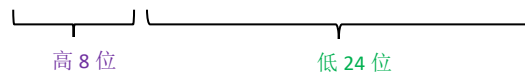
格式: **CNCAPushType**, <value>

其值是一个 32 位数 (以十进制显示), 高 8 位为 CNC 段类型, 低 24 位为本段 CNC 参与轴。

以上例中导出的 UserProgram 中第 2 段 (CCNCAPushType,16973823) 为例, 将其值转为 2 进制:

```
65 // Segment index: 2
66 //
67 // Push Linear motion, involved: A, C
68 // Go to: A=0, C=100000
69 // Velocity: 100000
70 // End Velocity: 100000
71 //
72 CCNCAPushType,16973823
73 CCNCAPushParam,0
74 CCNCAPushParam,100000
75 CCNCAPushParam,100000
76 CCNCAPushParam,100000
```

16973823 (D) ---> 0000 0001 0000 0010 1111 1111 1111 1111 (B) ---> 1 02FFFF (H)



1 02FFFF (H):

1:代表本 CNC 段类型为“1-Linear motion”, 和 2.2 章节 CNC 各 Segment ID 相对应;

02FFFF: 代表该 CNC 段关联轴, F 表示该轴不参与, 本 CNC 段关联轴为 A、C 轴, 因此为 0 和 2, “0”代表 A 轴, “1”代表 B 轴, “2”代表 C 轴...依次类推, 从小到大按顺序排列, CNC 最多可以关联 6 个轴, F 表示该位置无关联轴。

- ❖ **CNCAPushParam**: 用于定义推入 CNC FIFO 参数的值。

格式: **CNCAPushParam**, <Value>

其值是一个 32 位数 (以十进制显示), DSP 将会按照适合此段类型的正确顺序 (以及正确的参数数量) 推送参数到 CNC FIFO, 一旦最后一个参数被推送入 (控制器知道该段类型所需的参数数量), 控制器通过将“结束段定义” (与“开始段定义”相同) 推入到 CNC FIFO 中来关闭该段。

此时, 由于段已关闭, 下一个 CNCAPushParam 指令将会被拒绝并且会出现错误, 只有 CNCAPushType 有效。

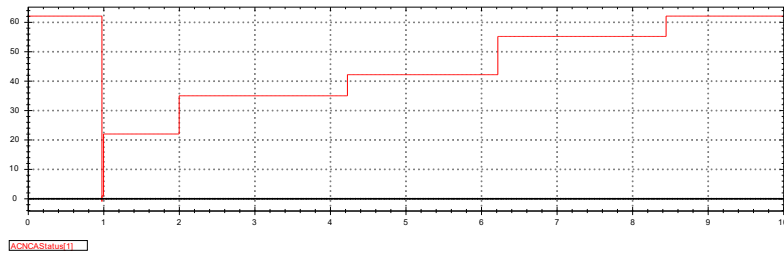
控制器会返回“OK>”或者错误代码 (如有错误发生的话), 如发生错误将, 则不会将任何内容推送到 CNC FIFO 中, 并保持此参数之前的状态。

推入最后一个参数后, 控制器还将验证该段的所有输入数据对于给定端类型是否有效 (参数是否在范围内、关联轴定义是否合理), 如果无效。则推送值将会被拒绝并且以相匹配的错误显示出来。

- 以下关键字用于管理 CNC FIFO 状态

- ❖ **CNCAStatus[ ]**: 用于描述 CNC FIFO 状态的只读数组, 某些状态在复位或重新上电后会失效, 某些状态在 CNC FIFO 为空时无效, 某些状态在 CNC FIFO 没有在运行是无效等等, 当这些状态无效时其值将会置为-1 (当这些状态有效时, 该值不是任何状态的有效值)。

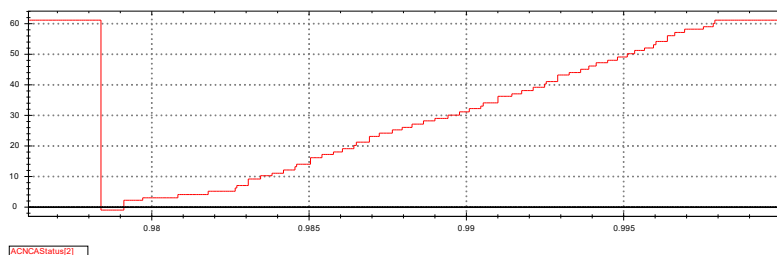
- **CNCStatus[1]** ---> CNC FIFO 中的最后一个参数的后一个索引



\*时间周期为 10s, 以 10%开始运动触发数据采集

- **CNCStatus[2]** ---> **LAST\_PUSHED\_ENTRY\_INDEX**

CNC FIFO[] (CNC FIFO 内存缓冲区) 的索引, 指向最近推送的项目;

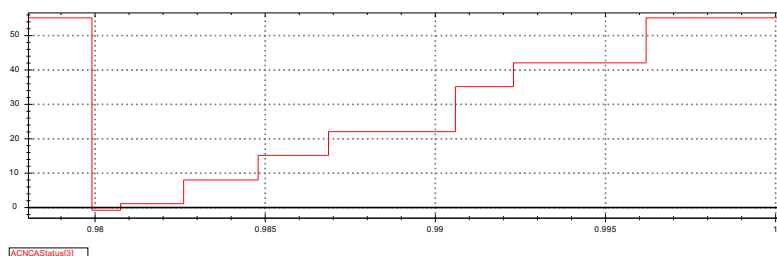


\*时间轴为运动开始之前

- **CNCStatus[3]** ---> **LAST\_SEGMENT\_ENTRY\_INDEX**

CNC FIFO[] (CNC FIFO 内存缓冲区) 的索引, 指向最后推送 (或正在推送) 段的开始。

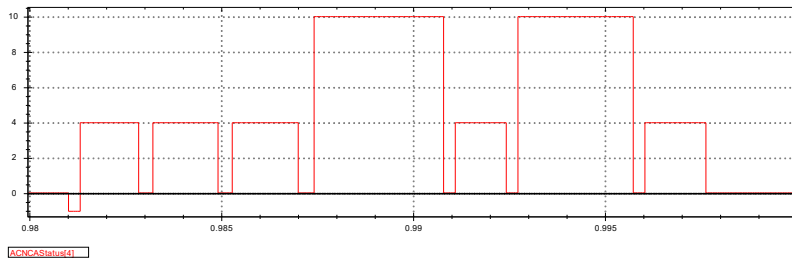
```
-----
ACNC FIF O[1-100]
1: 263>; 2: 100663295>; 3: 100>; 4: 10000000>; 5: 10000000>; 6: 0>; 7:
263>; 8: 519>; 9: 16973823>; 10: 0>; 11: 100000>; 12: 100000>; 13:
100000>; 14: 519>; 15: 775>; 16: 16973823>; 17: 100000>; 18:
100000>; 19: 100000>; 20: 100000>; 21: 775>; 22: 1037>; 23:
33751039>; 24: 100000>; 25: -100000>; 26: 0>; 27: 0>; 28: 1>; 29:
100000>; 30: 100000>; 31: 0>; 32: 785398>; 33: 5497786>; 34: 1037>;
35: 1287>; 36: 16973823>; 37: -100000>; 38: -100000>; 39: 100000>;
40: 100000>; 41: 1287>; 42: 1549>; 43: 33751039>; 44: -100000>; 45:
100000>; 46: 0>; 47: 0>; 48: 1>; 49: 100000>; 50: 100000>; 51: 0>; 52:
3926990>; 53: 8639379>; 54: 1549>; 55: 1799>; 56: 16973823>; 57: 0>;
58: 100000>; 59: 100000>; 60: 0>; 61: 1799>; 62: 0>; 63: 0>; 64: 0>; 65:
0>; 66: 0>; 67: 0>; 68: 0>; 69: 0>; 70: 0>; 71: 0>; 72: 0>; 73: 0>; 74: 0>;
75: 0>; 76: 0>; 77: 0>; 78: 0>; 79: 0>; 80: 0>; 81: 0>; 82: 0>; 83: 0>; 84:
0>; 85: 0>; 86: 0>; 87: 0>; 88: 0>; 89: 0>; 90: 0>; 91: 0>; 92: 0>; 93: 0>;
94: 0>; 95: 0>; 96: 0>; 97: 0>; 98: 0>; 99: 0>; 100: 0>
-----
```



\*时间轴为运动开始之前

- **CNCStatus[4]** ---> **LAST\_SEGMENT\_TOTAL\_NUM\_PARAMETERS\_INDEX**

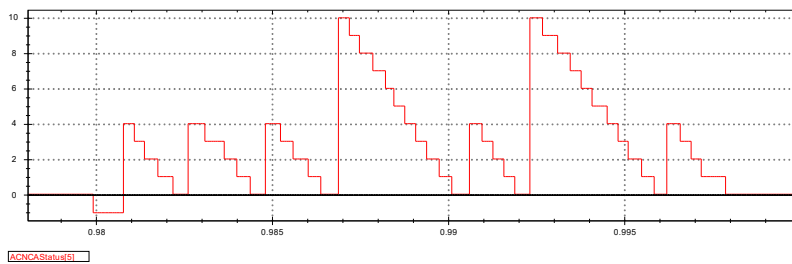
当前推送的 Segment 需要推送的参数 (CNCAPushParam) 总数, 如当前没有正在推送的 CNC 段, 则值为 0;



\*时间轴为运动开始之前

- **CNCStatus[5] ---> LAST\_SEGMENT\_NEED\_TO\_PUSH\_NUM\_PARAMETERS\_INDEX**

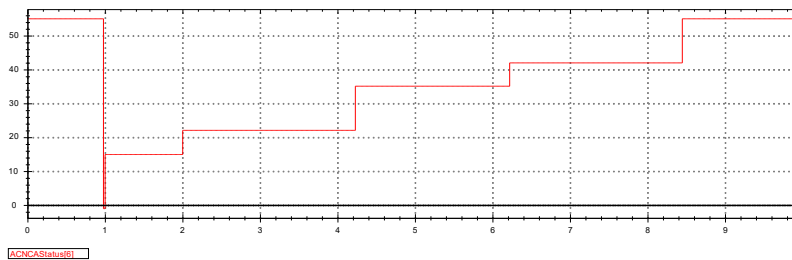
当前推送的 Segment 仍需要推送的参数 (CNCAPushParam) 数量, 如当前没有正在推送的段, 则为 0,



\*时间轴为运动开始之前

- **CNCStatus[6] ---> IN\_MOTION\_SEGMENT\_ENTRY\_INDEX**

CNCFIFO[] (CNC FIFO 内存缓冲区) 的索引, 指向当前执行段的开头 segment (CNC FIFO 运行期间有效);

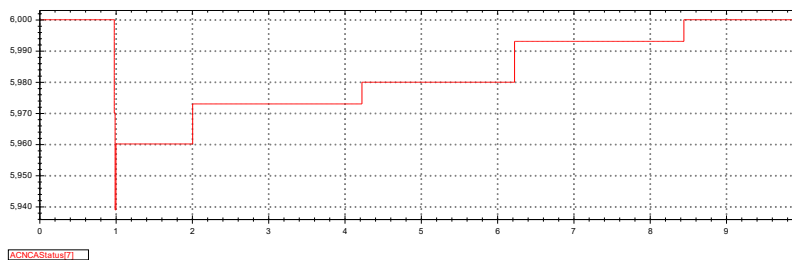


\*时间周期为 10s, 以 10%开始运动触发数据采集

- **CNCStatus[7] ---> FREE\_SPACE\_INDEX**

CNC FIFO 内的可用空间 (未使用的 32bit ENTRY),

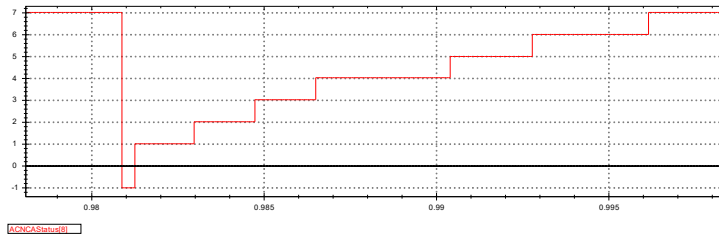
注意避免将新的 Segment 放置于 CNC FIFO 的最后 20 个 ENTRY 处 (有助于更有效地处理循环 CNC FIFO)



\*时间周期为 10s, 以 10%开始运动触发数据采集

- **CNCStatus[8] ---> LAST\_USED\_ID\_INDEX**

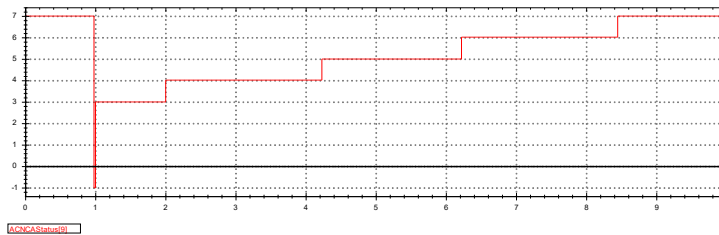
最后使用的 ID（或如果当前段已推送但尚未关闭，则为当前段使用的 ID）；



\*时间轴为运动开始之前

- [CNCStatus\[9\] ---> IN\\_MOTION\\_ID\\_INDEX](#)

运动期间，当前执行的 Segment 的 ID；



\*时间周期为 10s，以 10%开始运动触发数据采集

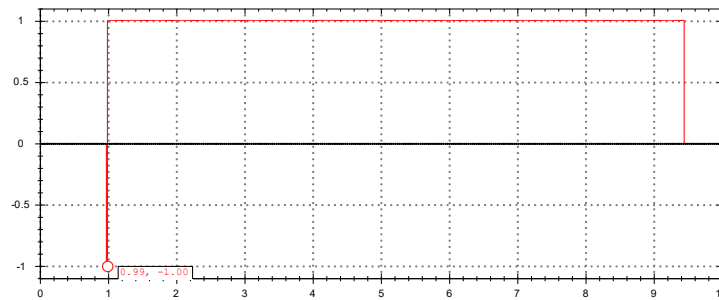
- [CNCStatus\[10\] ---> CNC\\_FIFO\\_MOTION\\_STATUS\\_INDEX](#)

Bit\_0: 当 CNCA 正在运行时值为“1”，否则为“0”；

Bit\_3: 当 CNCA 收到“Stop”停止请求时值为“1”，否则为“0”；

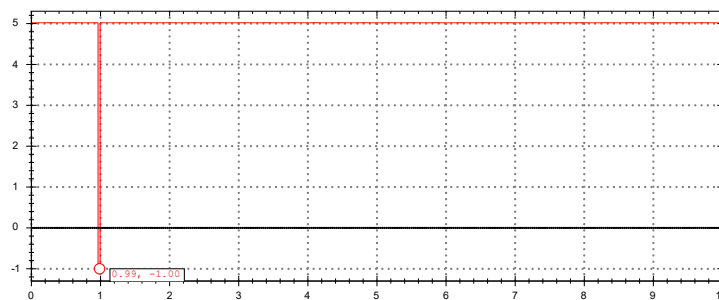
Bit\_12: 当 CNCA 收到“StopCNCA”停止 CNC 请求时值为“1”，否则为“0”；

其他 bit 保留暂未使用。



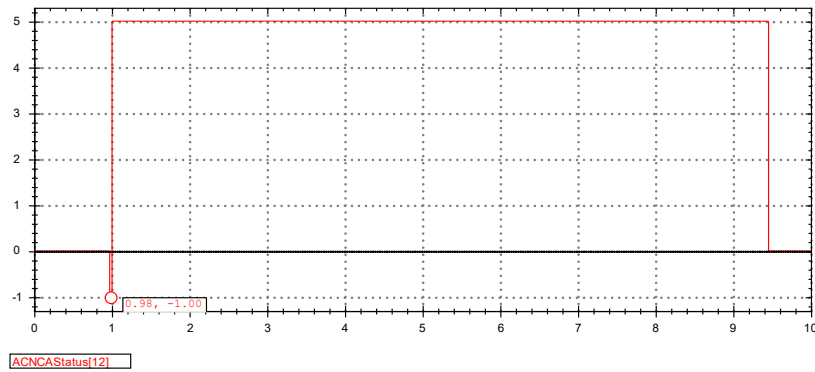
\*时间周期为 10s，以 10%开始运动触发数据采集

- [CNCStatus\[11\] ---> CNC\\_STATUS\\_CNC\\_FIFO\\_MEMBERS\\_AXES\\_INDEX](#)



\*时间周期为 10s，以 10%开始运动触发数据采集

- [CNCAStatus\[12\]](#) ---> [CNC\\_FIFO\\_INVOLVED\\_AXES\\_INDEX](#)



\*时间周期为 10s，以 10%开始运动触发数据采集

- [CNCAStatus\[13\]](#) ---> [CNC\\_STATUS\\_MAX\\_ACCELERATION\\_COUNTER\\_INDEX](#)  
由于轴加速度限制，倒角速度必须降低，[CNCAStatus\[13\]](#)用于对该自动倒角 segment 进行计数，在上电、reset 复位和 [CNCAClear](#) 指令是，该值将会被清 0；
  - [CNCAStatus\[14\]](#) ---> [CNC\\_STATUS\\_MAX\\_VELOCITY\\_JUMP\\_COUNTER\\_INDEX](#)  
当其中某一关联轴速度跳跃超过最大速度跳跃时，线性运动的前一个 segment 的计数速度将会被降低，[CNCAStatus\[14\]](#)将会对此进行计数，在上电、reset 复位和 [CNCAClear](#) 指令时，该值将会被清 0；
- ❖ **CNCAClear:** 用于清除和重置整个 CNCA FIFO 的 function 类关键字，运动期间该关键字禁止使用；

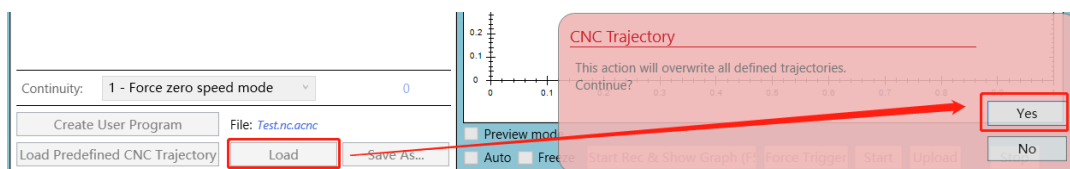
### 2.3.3 示例 3——导入 G/M-Code

Agito CNC 支持 G-Code 和 M-Code 导入生成 CNC segment。

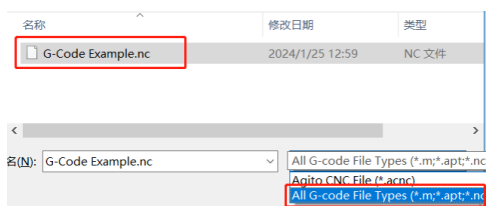
以下以一个简单的 G-Code 文件作为示例：

```
G-Code Example.nc
1 G0 X100.000 Y200.000 F30.000
2 G1 X200.000 Y300.000
3 M02
```

- 第 1 步：在 [MOTION](#) ---> [CNC](#) 界面，点击“Load”弹出右侧提示窗，点“[Yes](#)”弹出文件选择窗；



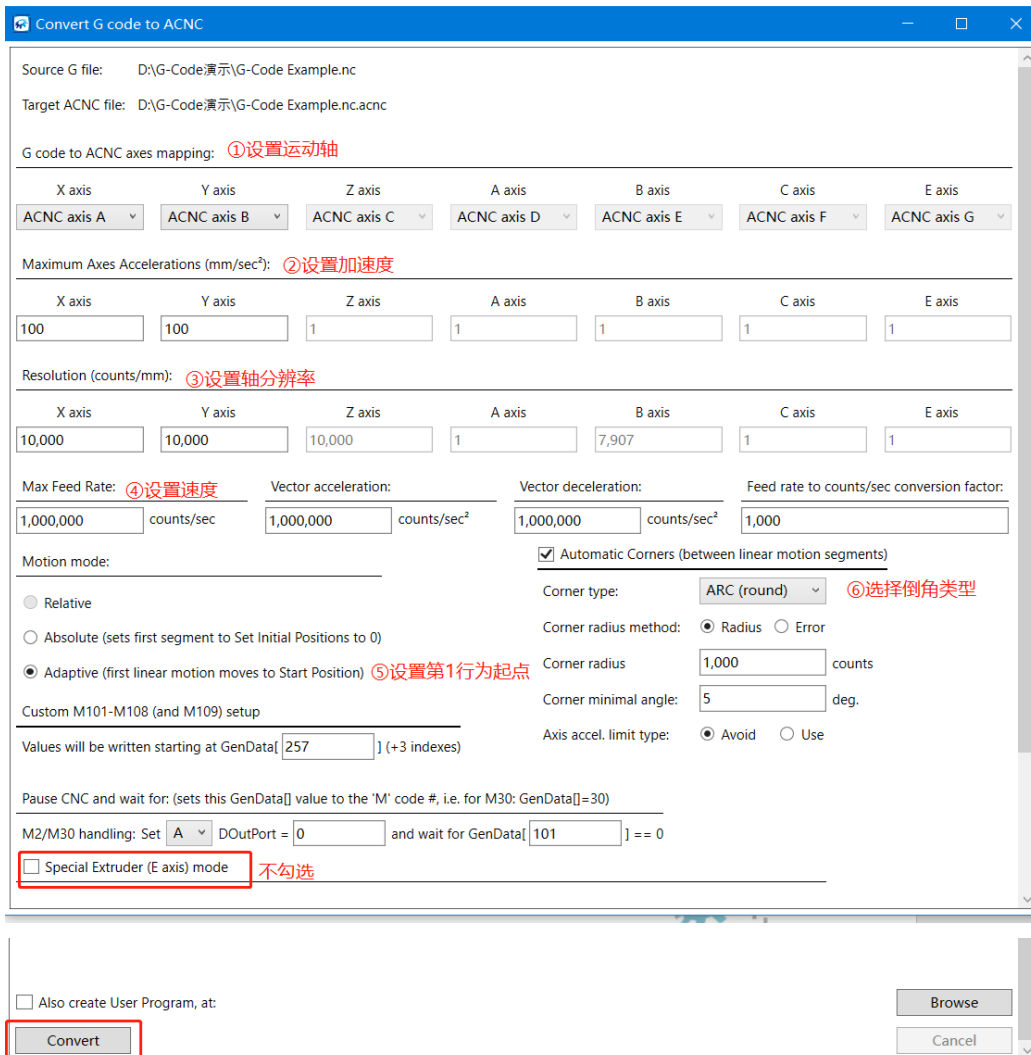
- 第 2 步：在弹出的文件选择窗中进入本地 G-Code 文件路径选择对应文件



支持以下文件格式：

All G-code File Types (\*.m;\*.apt;\*.nc;\*.cnc;\*.ncc;\*.ecs;\*.cnc;\*.tap;\*.NC;\*.NCC;\*.fnc;\*.dnc;\*.ncg;\*.gc;\*.fan;\*.fgc;\*.din;\*.txt;\*.xpi;\*.hnc;\*.h;\*.i;\*.ncp;\*.min;\*.GCD;\*.rol;\*.mpr;\*.ply;\*.OUT;\*.eia;\*.plt;\*.gcd;\*.ncd;\*.sbp;\*.mpf)

- 第 3 步：从上到下分别设置运动轴、加速度、分辨率、速度、倒角类型等参数，然后点击“Convert”进行转换，此界面也可以将 G-Code 生成 User Program（勾选 Also create User Program at:）



Convert G code to ACNC

Source G file: D:\G-Code演示\G-Code Example.nc  
Target ACNC file: D:\G-Code演示\G-Code Example.nc.acnc

G code to ACNC axes mapping: ①设置运动轴

X axis	Y axis	Z axis	A axis	B axis	C axis	E axis
ACNC axis A	ACNC axis B	ACNC axis C	ACNC axis D	ACNC axis E	ACNC axis F	ACNC axis G

Maximum Axes Accelerations (mm/sec<sup>2</sup>): ②设置加速度

X axis	Y axis	Z axis	A axis	B axis	C axis	E axis
100	100	1	1	1	1	1

Resolution (counts/mm): ③设置轴分辨率

X axis	Y axis	Z axis	A axis	B axis	C axis	E axis
10,000	10,000	10,000	1	7,907	1	1

Max Feed Rate: ④设置速度    Vector acceleration:    Vector deceleration:    Feed rate to counts/sec conversion factor:

1,000,000	counts/sec	1,000,000	counts/sec <sup>2</sup>	1,000,000	counts/sec <sup>2</sup>	1,000
-----------	------------	-----------	-------------------------	-----------	-------------------------	-------

Motion mode:

Relative

Absolute (sets first segment to Set Initial Positions to 0)

Adaptive (first linear motion moves to Start Position) ⑤设置第1行为起点

Custom M101-M108 (and M109) setup

Values will be written starting at GenData[ 257 ] (+3 indexes)

Pause CNC and wait for: (sets this GenData[] value to the 'M' code #, i.e. for M30: GenData[]=30)

M2/M30 handling: Set [ A ] DOutPort = 0 and wait for GenData[ 101 ] == 0

Special Extruder (E axis) mode 不勾选

Also create User Program, at:    Browse    Cancel

- 第 4 步：上一步转换成功后将会生成 CNC Segment;

Trajectory	
CNC Trajectory:	Valid
1 - Vector Params.:	100%, 1000000, 1000000, 0.000 msec
2 - Corner Params.:	ARC, radius, 1000, 5 deg., avoid limit
3 - Linear motion:	A=1000000, B=2000000
4 - Linear motion:	A=2000000, B=3000000
5 - Write DOutPort:	ADOutPort=0

