



# Agito Gear & ECAM 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	2024/2/19

※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，如需最新本档文档，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	Gear 和 ECAM 简介	4
2	齿轮（Gear）使用方法	5
2.1	在 PCSuite 界面操作齿轮	5
2.2	在开启 ModRev（编码器模量）时使用齿轮	7
2.3	在 IDE 编程环境中操作齿轮	8
3	凸轮（ECAM）使用方法	10
3.1	在 PCSuite 中认识 ECAM	10
3.2	通过 PCSuite 开启 ECAM	13
4	常见问题解答（FAQ）	15

# 1 介绍

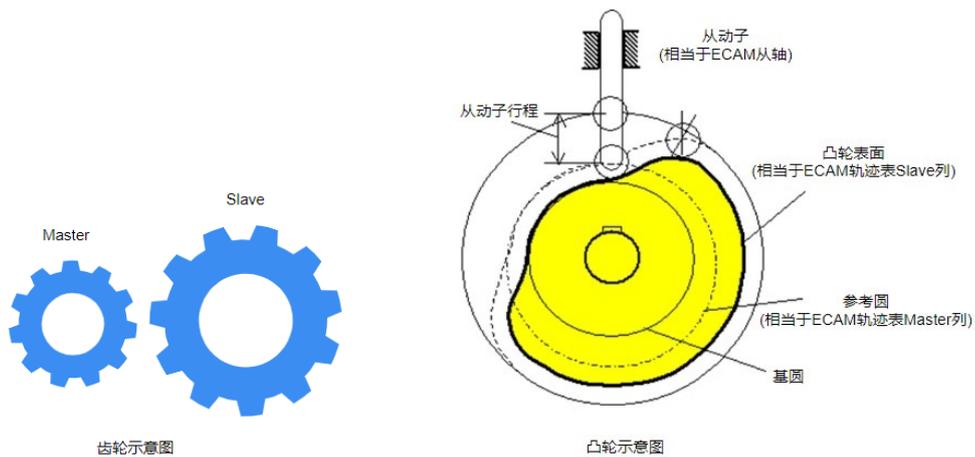
## 1.1 关于手册

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 齿轮（Gear）和凸轮（ECAM）功能使用方法，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 Gear 和 ECAM 简介

齿轮（Gear）和电子凸轮（ECAM）都是一种主从跟随模式，通常是从轴根据主轴位置按照一定的关系进行运动。不同的是齿轮是按照固定的比例进行实时跟随，而 ECAM 根据预定义的样条曲线或查表方式进行跟随。



值得注意的是：

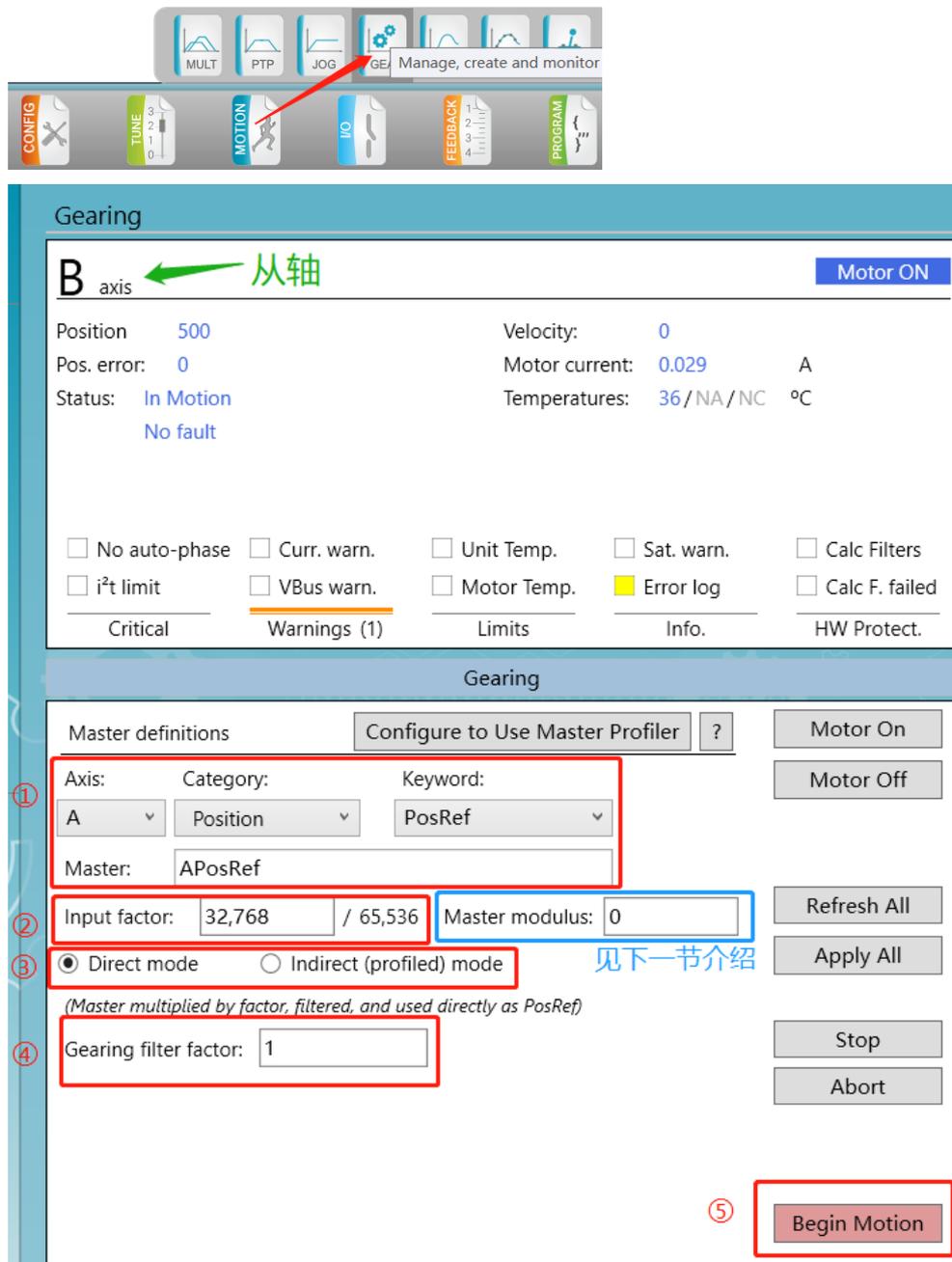
- 无论是 Gear 还是 ECAM 模式都只需要设置从轴参数，即只需要开启从轴为 Gear 或 ECAM 运动模式（MotionMode）；
- 主轴无需设置任何参数，但因为是主从跟随模式（齿轮为直接跟随模式），从轴的 motion profile 是在主轴的运动指令（如速度、加减速等）上乘一个齿轮比系数，因此从轴在开启跟随模式之前，需要确保从轴可以按照主轴运动参数乘齿轮比之后的参数正常进行运动，否则从轴将可能出现报警、停止运动等异常发生；
- 在 Gear+ModRev 使用条件下，需要避免将齿轮比设置为（即 MasterFact）设为负数，如用户希望从轴的运动方向与主轴运动方向相反，可以将从轴的编码器和电流方向同时取反（在 AuotoPhasing 中的 direction 界面），取反之后编码器的正向就会和原来相反，效果等同于齿轮比为负数。
- 在 InMotion 状态下无法修改齿轮或凸轮的相关参数，如需更改需要先停止当前运动（发送 Stop 或 Abort 指令）。

## 2 齿轮（Gear）使用方法

Agito 控制器支持自定义变量跟随，不限于位置，但通常使用主轴的 PosRef 进行齿轮跟随。以下示例以 A 轴为主轴（Master），B 轴为从轴（Slave），2:1 齿轮比跟随运动作为演示。

### 2.1 在 PCSuite 界面操作齿轮

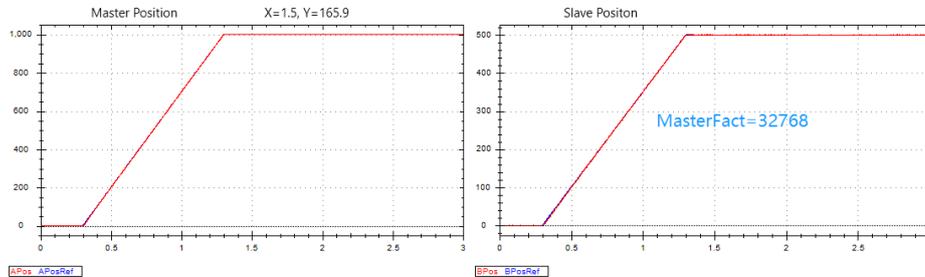
- 第 1 步：切换菜单栏到**从轴**齿轮设置界面；



- 第 2 步：按以上操作步骤开启从轴齿轮模式，值得注意的是当点击“Begin Motion”之后从轴将会处于“In Motion”运动状态，即使没有实际运动；

注：以下括号内为该参数关键字

- ① **Master (GearMaster)**: 用于设置跟随指令源，通常设置为主轴位置指令，即 PosRef（本例中主轴为 A 轴，所以设为 APosRef），当然如用户有特殊应用，也可以选择其他指令进行关联，②中的“GearMaster 对应值”指的是对应关键字的实时值；
- ② **Input factor (MasterFact)**: 设定主从轴**齿轮比**， $\frac{\text{GearMaster 对应值}}{B\text{PosRef}} = \frac{\text{MasterFact}}{65536}$ ，GearMaster 即①中的指令，如主从 1:1 跟随，则该系数为 65536，如 2:1 跟随则为 32768，依次类推；



- ③ **Direct/Indirect mode (MotionMode=5/6)**: 选择直接齿轮耦合或自定义轨迹耦合；
  - ◆ **方式 1**: 直接耦合 (MotionMode=5)，从轴按照主轴的轨迹规划进行跟随，在无特殊应用的情况下，通常使用直接耦合方式；

值得注意的是：由于底层对 motion profile 的计算是按顺序执行的，先是 A，接着是 B，然后是 C，因此当需要严格跟随时，请将 A 轴作为控制主轴，B 轴作为从轴 (Slave)，如将 B 作为主轴，A 做从轴，其之间将会有 61us 的延迟；

Master definitions Configure to Use Master Profiler ?

Axis:  Category:  Keyword:

Master:

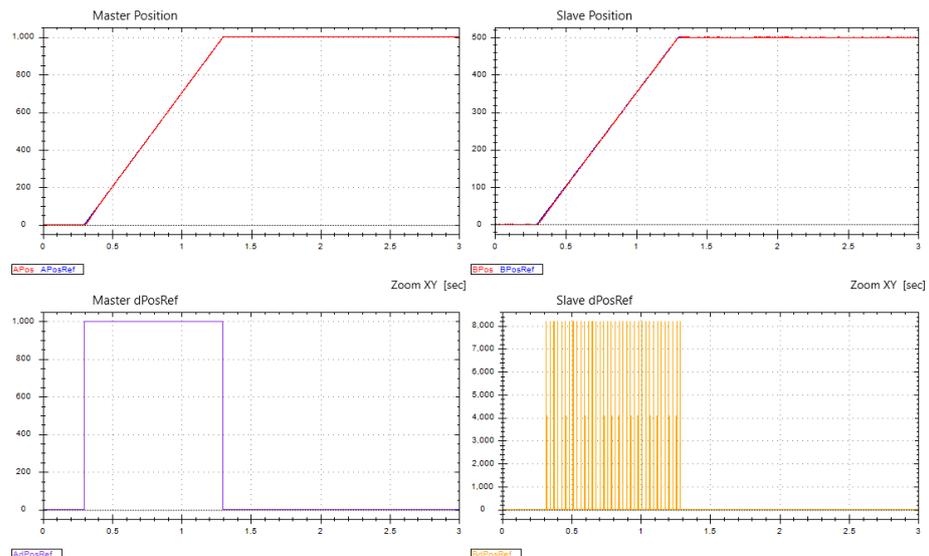
Input factor:  / 65,536 Master modulus:

Direct mode  Indirect (profiled) mode

(Master multiplied by factor, filtered, and used directly as PosRef)

Gearing filter factor:

参数设定:



运动图像:

- ◆ **方式 2**: 间接耦合 (MotionMode=6)，从轴根据**主轴绝对运动距离和齿轮比的乘积**作为绝对运动距离，并按照用户自定义的速度、加/减速度等参数进行运动，
- 值得注意的是：间接耦合方式中从轴的 profile 速度**不能超过主轴速度**，否则速度将会按照**主轴速度\*齿轮比**进行运动；

Master definitions Configure to Use Master Profiler ?

Axis: A Category: Position Keyword: PosRef

Master: APosRef

Input factor: 32,768 / 65,536 Master modulus: 0

Direct mode  Indirect (profiled) mode

*(Master multiplied by factor, and used as profiler absolute target)*

Acceleration: 30,000

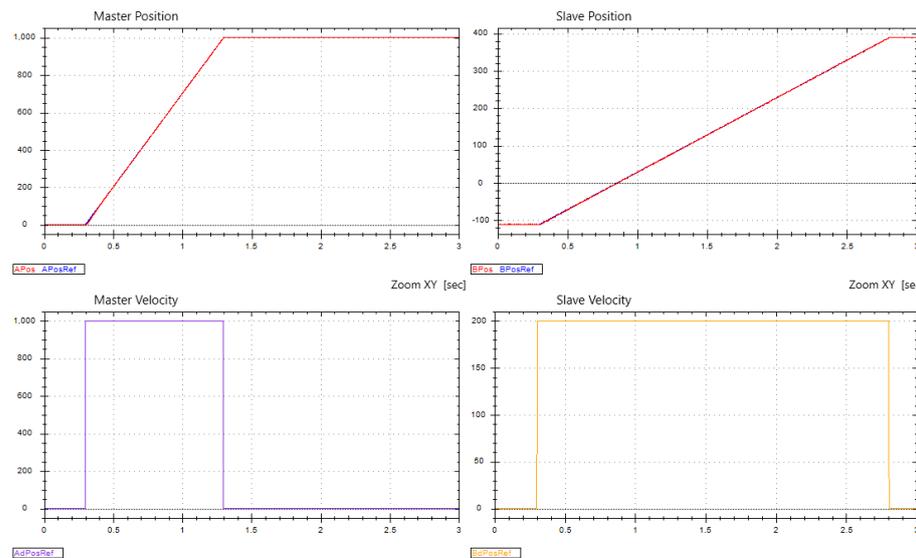
Deceleration: 30,000 X 1 user-units/sec<sup>2</sup>

Emergency decel.: 30,000

Smooth: (msec) 0.000

Speed: 200 user-units/sec

参数设定:



运动图像:

- ④ **Gearing filter factor (MasterFilt)**: 滤波通常用于防止位置指令出现大的跳跃，特别是当 MaterFact 很大时，其值在 1~64 之间，通常设为 1，即严格完全跟随；  

$$\text{PosRefK} = (\text{MasterPosK} * \text{MasterFilt} + \text{PosRefK-1} * (64 - \text{MasterFilt})) / 64$$
- ⑤ **BeginMotion**: 点击后从轴将会开启跟随模式，运动状态将会处于 “InMotion” 状态，直至用户发送 “Stop” 或 “Abort” 停止指令；

## 2.2 在开启 ModRev（编码器模量）时使用齿轮

在某些应用中，旋转电机需要向一个方向无限运动，此时将会用到 ModRev 功能（此功能在 Feedbacks 界面设置，本文将不再赘述），但用户又希望在无限运动模式下使用齿轮模式，则 MasterModRev 需要按照以下规则设定相应参数，其他参数设定参阅 2.1 章节介绍：

假设 A 轴为主轴（Master），B 轴为从轴（Slave），且 AModRev=a，BModRev=b，齿轮比为 1:5，则 MasterModRev 设置为：AMasterModRev=0，BMasterModRev=a\*5，依次类推；

Gearing

Master definitions Configure to Use Master Profiler ?

Axis: A Category: Position Keyword: PosRef

Master: APosRef

Input factor: 327,680 / 65,536 Master modulus: 50,000

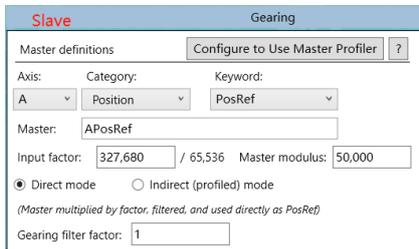
Direct mode  Indirect (profiled) mode

*(Master multiplied by factor, filtered, and used directly as PosRef)*

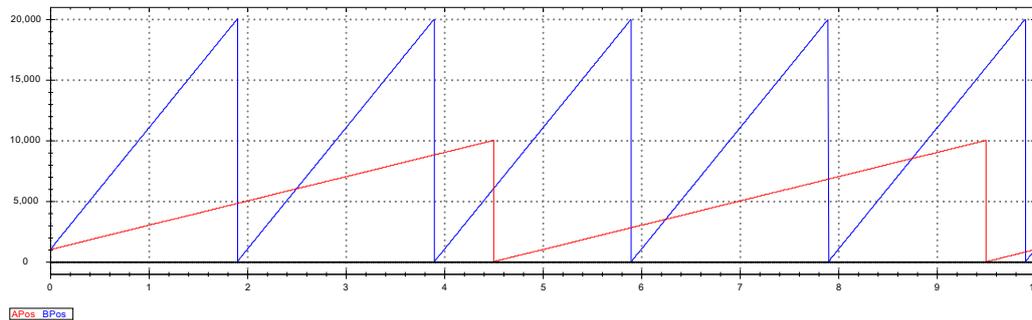
Gearing filter factor: 1

注：不使用 ModRev 功能时请将主从轴 MasterModRev 都设为 0.

例如本例中：AModRev=10000，BModRev=20000，齿轮比为 1:5，则 B 轴（Slave）的 Input factor 分子系数 (即 BMasterFact) 设为  $65536 * 5 = 327680$ ，B 轴 Master modulus (即 BMasterModRev) 设为  $AModRev * 5 = 50000$ ；



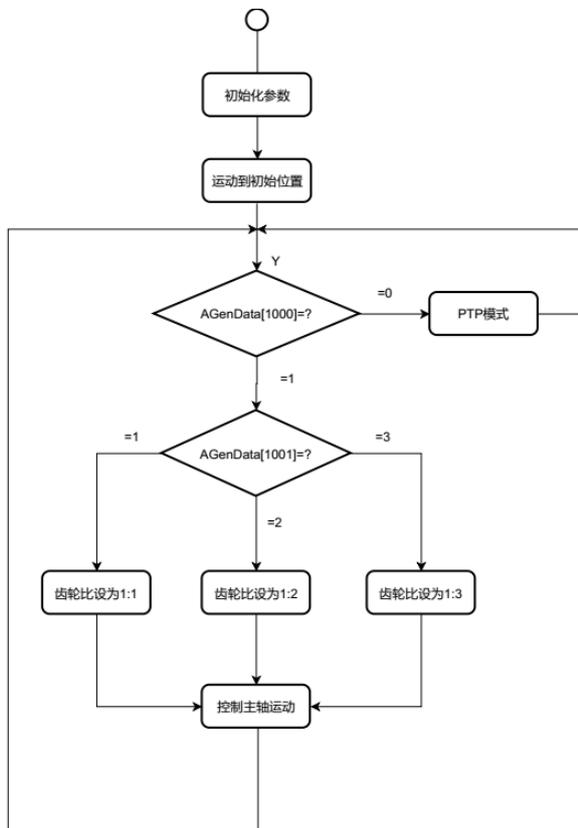
运动图像：



## 2.3 在 IDE 编程环境中操作齿轮

以下以一段在 IDE 环境中实现齿轮比切换作为演示：

程序逻辑框图：



IDE 程序:

```

14 #define var GearTemp //在.puh2 中定义一个变量, 用于存放当前齿轮比系数

22 ///////////////////////////////////////////////////////////////////
23 //初始化变量
24 AGenData[1000]=0
25 AGenData[1001]=1
26 AGenData[1002]=0
27 GearTemp=0
28
29 //运动到起始位置
30 AMotionMode=1
31 BMotionMode=1
32 ARelTrgt=0
33 AAbsTrgt=1000
34 ASpeed=5000
35 AAccel=50000
36 ADecel=50000
37 BRelTrgt=0
38 BAbsTrgt=2000
39 BSpeed=5000
40 BAccel=50000
41 BDecel=50000
42 AMotorOn=1
43 BMotorOn=1
44 ABegin
45 BBegin
46
47 //判断A、B轴运动到位
48 while ((AInTargetStat!=4) || (BInTargetStat!=4))
49 end
50
51 while ((AConFlt==0) && (BConFlt==0))
52     if (AGenData[1000]==0)
53         if (BMotionMode==5)
54             BStop
55             while (BMotionStat!=0)
56             end
57             GearTemp=0
58             BMotionMode=1
59         end
60     else if (AGenData[1000]==1)
61         if (GearTemp!=AGenData[1001])
62             switch (AGenData[1001])
63                 case 1
64                     BStop
65                     while (BMotionStat!=0)
66                     end
67                     BMotionMode=5
68                     BMasterFact=65536
69                     GearTemp=AGenData[1001]
70                     BBegin
71                     break
72
73                 case 2
74                     BStop
75                     while (BMotionStat!=0)
76                     end
77                     BMotionMode=5
78                     BMasterFact=131072
79                     GearTemp=AGenData[1001]
80                     BBegin
81                     break
82                 case 3
83                     BStop
84                     while (BMotionStat!=0)
85                     end
86                     BMotionMode=5
87                     BMasterFact=196608
88                     GearTemp=AGenData[1001]
89                     BBegin
90                     break
91             end
92         end
93     end
94 ///////////////////////////////////////////////////////////////////

```

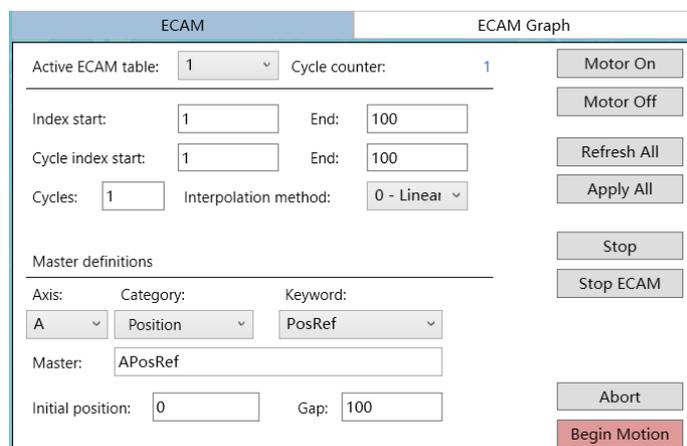
### 3 凸轮（ECAM）使用方法

凸轮和齿轮最大的区别是凸轮通常用于时变的位置跟随，其运动位置会根据主轴的运动位置以及用户定义的跟随轨迹实时变化。Agito 电子凸轮模式支持单次\多次和无限次的循环，可以按照混合轨迹或特定轨迹加速进入或减速脱离 ECAM 循环，可以帮助用户灵活定义多轴联动。

#### 3.1 在 PCSuite 中认识 ECAM

PCSuite 上设置 ECAM 参数主要涉及 2 个界面：

**界面一：** 切换菜单栏到**从轴** ECAM 界面：MOTION→ECAM



按钮（【】内为发送命令）：

- Stop: 【Stop】停止当前轴运动；
- Stop ECAM: 【StopECAM】停止凸轮跟随；
- Begin Motion: 【MotionMode=7 → MotorOn=1 → Begin】，开始进行凸轮跟随；

**参数：**

- ①Active ECAM table (ECAMTableNum): 选择当前 ECAM 运动使用的表格 x，注意目前只支持使用 1 个表，即 x 只能等于 1；

- ②Cycle counter (ECAMCycCount[1]): 用于记录每个 ECAM 表格当前的循环次数；

- ③Index start (ECAMStart[1]): 用于定义 ECAM 表第 1 个元素的索引（注：ECAM 表中的元素保存在 GenData[]数组中），

例如：表 1 的元素是从 GenData[1]~GenData[101]，则 ECAMStart[1]=1；

End (ECAMEnd[1]): 用于定义 ECAM 表的最后个元素的索引（GenData[]索引），

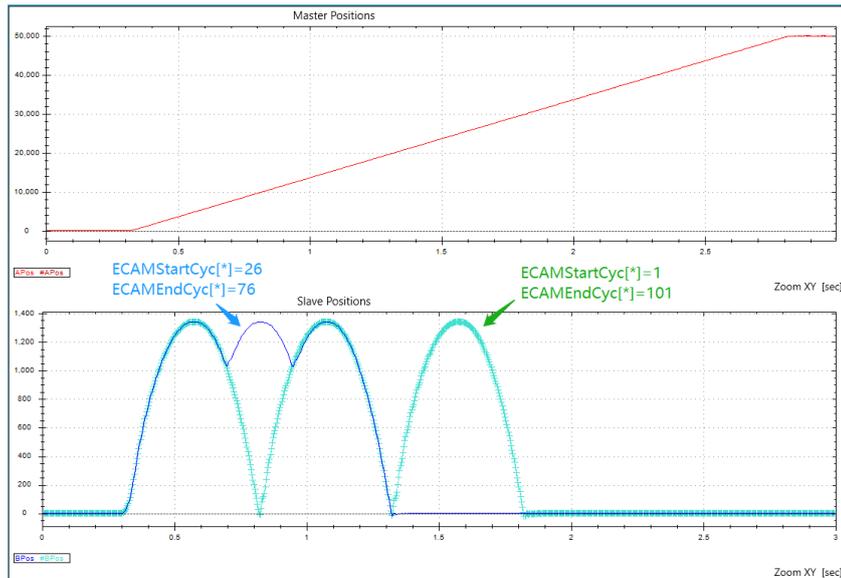
如本例中 ECAMEnd[1]=101；

- ④Cycle index start (ECAMStartCyc[1]): 用于定义 ECAM 表开始循环重复元素的索引，即循环的第 1 个索引，

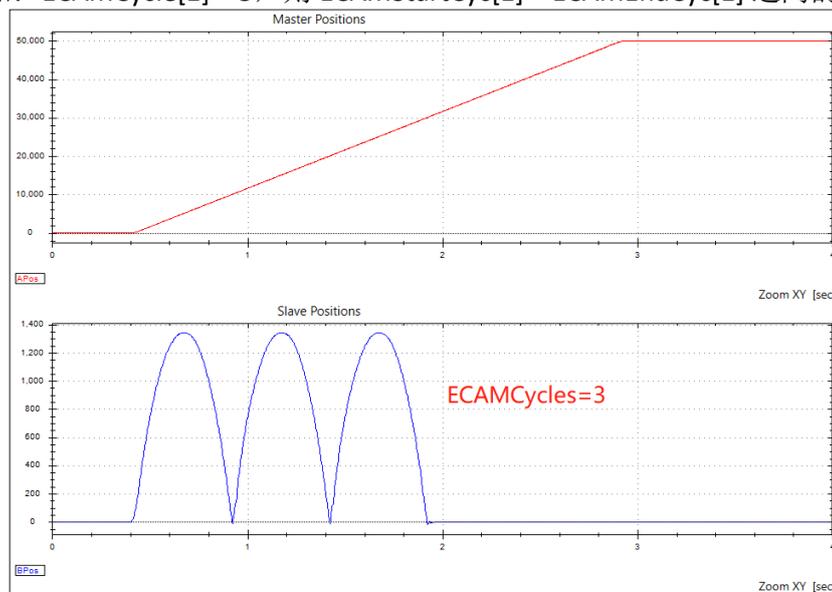
例如 ECAM 表 1 的元素是从 GenData[1]~GneData[101]，并且表中 GenData[26]~GenData[76]之间的元素需要循环重复，则 ECAMStartCys[1]=26；

End (ECAMEndCyc[1]): 用于定义 ECAM 表中循环重复的最后元素的索引，

例如本例中 ECAMEndCyc[3]=76；



- ⑤ **Cycles (ECAMCycles[1]):** ECAMCycles[1]用于定义 ECAM 表 1 运动循环重复的次数，  
例如：ECAMCycle[1] = 3，则 ECAMStartCyc[1] ~ ECAMEndCyc[1] 之间的部分将重复 3 次，



当 ECAMCycles 为正值时，在以循环起点对称的负方向上，向负方向运动 ECAM 将不会生效，ECAMCycles 可以设置为负值，表示运动以循环起点对称，并且将会在负方向和正方向上重复相同的次数，表中超出循环部分边界的部分通常用于加速或减速，如果 ECAMCycles 为负值，则到达负值最大位置后将会移动到新的逻辑位置，

无限 ECAM:

若 ECAMCycles[1] 值设置为 2147483647，控制器将执行无限次的 ECAM 循环，直到某些外部事件停止运动（例如：收到 Stop 或 StopECAM 命令，或触及限位开关，或电机下使能，或某些故障等），

负值则为-2147483648 来创建无限运动，

**Interpolation method (ECAMInterp[1]):** 该参数作为未来开发保留关键字，目前只能设为 0;

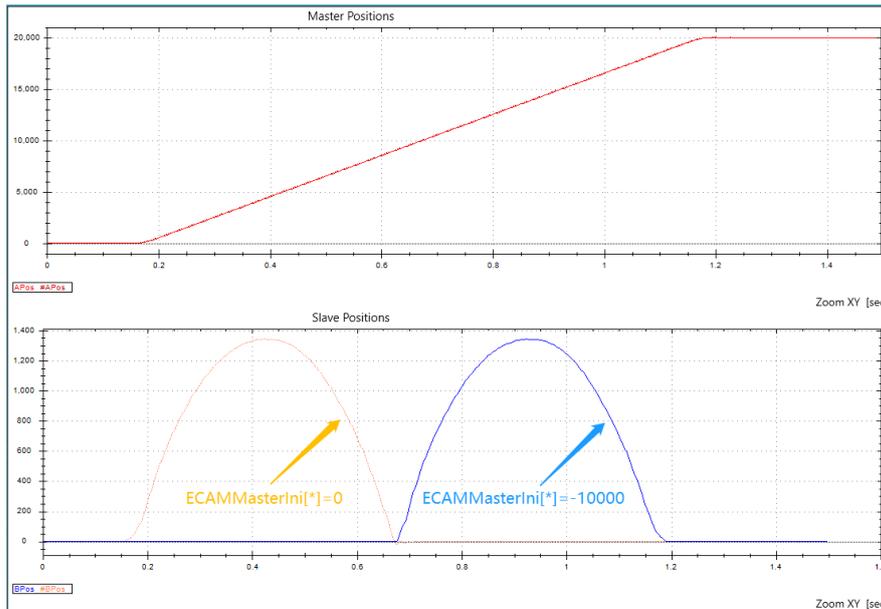
- ⑥ **Master (ECAMMaster[1]):** 用于定义 ECAM 主轴值的 CANCode，任何关键字（包括数组）都可以作为被跟随对象，但通常使用主轴 PosRef 作为主值，如 A 轴为主轴，则 ECAMMaster[1]=24，最简单的方法是通过 PCSuite 自动计算得到；

- ⑦ Initial position (ECAMMasterIni[1]): 如果需要补偿主轴运动, 则从起点位置中减去 ECAMMasterIni[x] 的值,

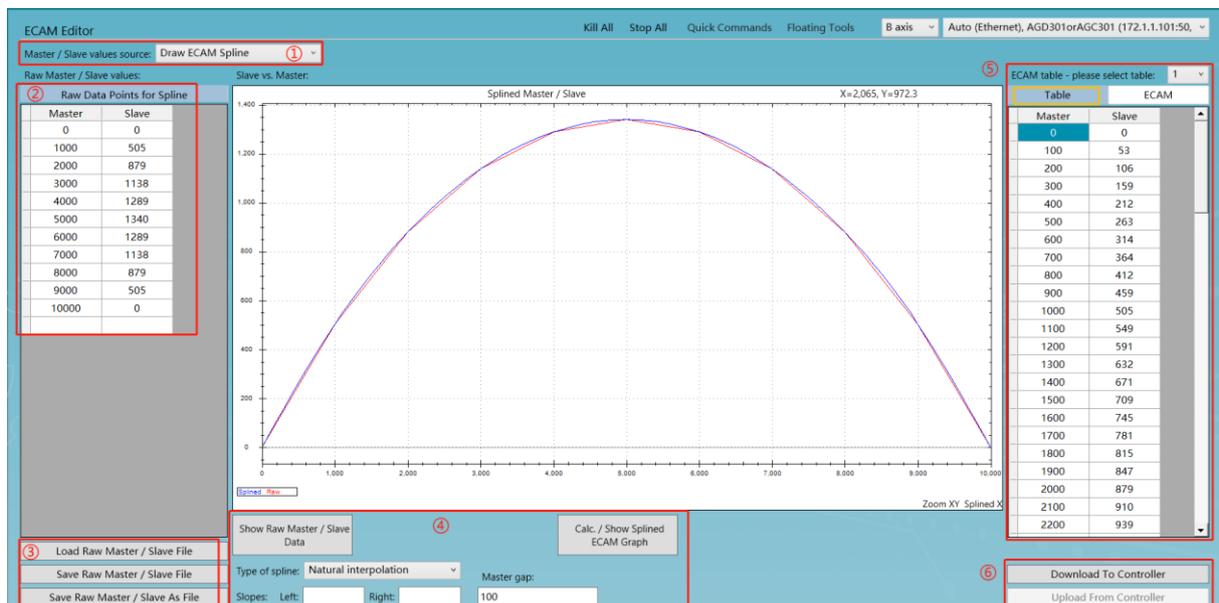
例如: 当 ECAMMasterIni[1]=0 且主轴位置为 0 时接收到开始命令, 则所有主位置都将相对于 0 位置, 向 ECAMStart[1] 指向的位置发出位置命令立即开始 ECAM 运动,

当 ECAMMasterIni[1]=-10000 并且主轴位置为 0 时接收到开始命令, 则所有主控位置都相对于 10000, 向 ECAMStart[1] 指向的位置发出位置命令立即开始 ECAM 运动;

Gap (ECAMGap[1]): 用于定于 ECAM 表中主轴每个点之间的间距, ECAMGap 为负值可用于反转主轴读数的方向;



界面二: 切换菜单栏到**从轴** ECAM 表编辑界面: DRAW→ECAM Editor

ECAM Editor

Master / Slave values source: Draw ECAM Spline

Raw Master / Slave values:

Master	Slave
0	0
1000	505
2000	879
3000	1138
4000	1289
5000	1340
6000	1289
7000	1138
8000	879
9000	505
10000	0

Splined Master / Slave

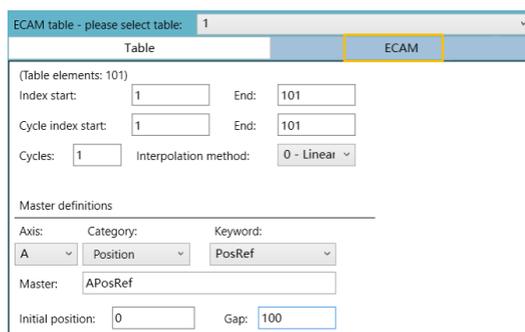
ECAM table - please select table: 1

Table	Master	Slave
1	0	0
1	100	53
1	200	106
1	300	159
1	400	212
1	500	263
1	600	314
1	700	364
1	800	412
1	900	459
1	1000	505
1	1100	549
1	1200	591
1	1300	632
1	1400	671
1	1500	709
1	1600	745
1	1700	781
1	1800	815
1	1900	847
1	2000	879
1	2100	910
1	2200	939

Download To Controller

Upload From Controller

- ① **Master/Slave values source**: 主从值跟随数据源，可以选择绘制轨迹（Draw ECAM Spline）或者查询 ECAM 表（ECAM table handling）；
- ② **Raw Data Points for Spline**: 编辑原始数据点位，Mater 为主轴相对位移，Slave 为从轴相对位移，注意此处的位移都是相应于从轴开启 ECAM 时主从轴位置的位移；
- ③ **Load Raw Master / Slave File**: 从本地导入保存在本地的\*.msf 格式 ECAM 表，  
**Save Raw Master / Slave File**: 保存 ECAM 表到控制器，  
**Save Raw Master / Slave As File**: 保存 ECAM 表到本地文件（\*.msf 格式）；
- ④ **Show Raw Master / Slave Data**: 点击按钮后将在图像中拟合出 ECAM 表的原始点轨迹，  
**Calc./ Show Splined ECAM Graph**: 点击按钮后将会根据设定的间距（Master gap）计算并拟合出样条曲线并显示的图像中；  
**Type of spline**: 选择“Natural interpolation”，  
**Master Gap**: 样条曲线间距，理论上间距细分的越小越接近真实轨迹，但值得注意的是：ECAM 表中元素总量受 GenData 数组大小限制（不同型号控制器有差异，如 AGD301 最大为 5000 个），同时在 IDE user program 中自定义变量会自动分配到一个 GenData，因此使用时需要注意 ECAM 表的大小；
- ⑤ **ECAM table**: 选择 ECAM 表，目前只能选择“1”，表中将会展示出④计算出的拟合点数据表；
- ⑥ **Download To Controller**: 将 ECAM 表下载到控制器中，每次修改 ECAM 表之后请点击此按钮保存数据到控制器才能生效；  
**Upload From Controller**: 选择①中的“ECAM table handling”并点击“Upload From controller”可以将上传读取当前保存在控制器中 ECAM 表元素数据；
- ⑦ **ECAM 参数**: 此界面和界面一中的参数部分完全相同，此处将不再赘述；



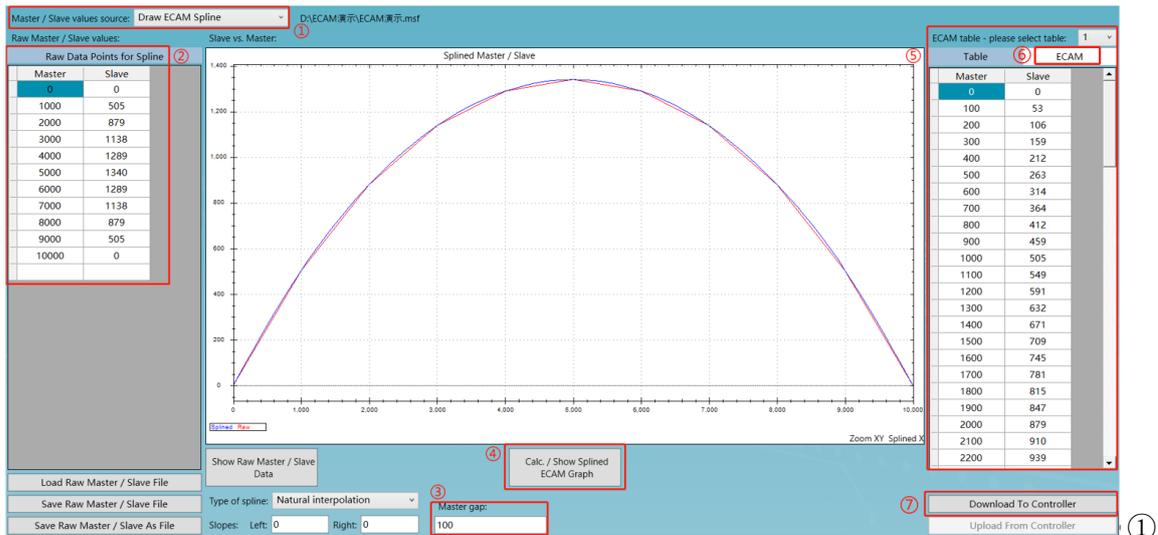
## 3.2 通过 PCSuite 开启 ECAM

以下以 A 轴作为 ECAM 主轴，B 轴为从轴，以一段圆弧为跟随轨迹作为示例。

- 第 1 步: 切换菜单栏到**从轴** ECAM 表编辑界面: DRAW→ECAM Editor



■ **第 2 步：**按以下步骤描绘 ECAM 样条曲线



选择绘制轨迹（Draw ECAM Spline）；

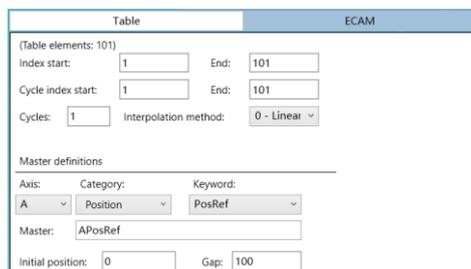
②根据用户的应用需求填入原始数据点位，Master 为主轴相对位移，Slave 为从轴相对位移，值得注意的是 Master 的第 1 个点位必须填“0”；

③根据用户需求填入样条曲线间距，注意数量的限制；

④点击“Calc./ Show Splined ECAM Graph”后将会根据设定的间距（Master gap）计算并拟合出样条曲线并显示的图像中，同时计算后的 ECAM 表元素将会展示在⑤中；

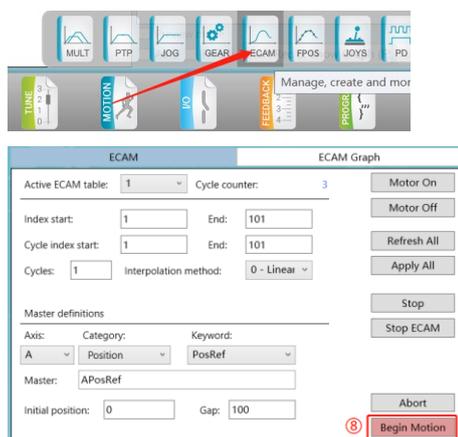
⑤用户可以对拟合出来的 Slave 中元素进行修改；

⑥点击“ECAM”，填入 ECAM 相关参数；



⑦点击“Download To Controller”将 ECAM 表及参数保存到控制器中；

■ **第 3 步：**切换菜单栏到从轴 ECAM 界面：MOTION→ECAM，点击⑧“Begin Motion”，从轴将开始和主轴进行跟随运动；

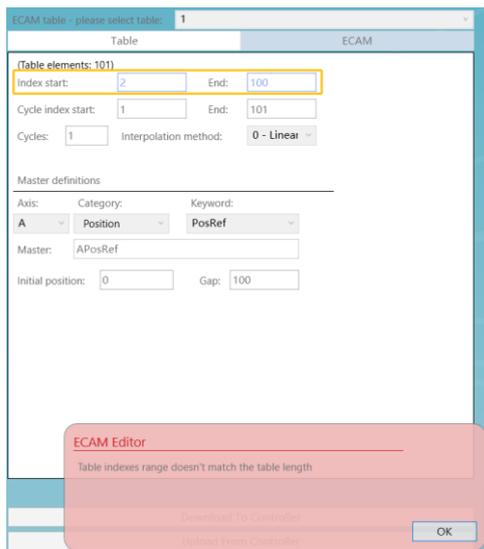


## 4 常见问题解答 (FAQ)

- **问题 1:** 填写完 ECAM 所有参数后点击“Download To Controller”，为什么会提示“Table indexes range doesn't match the table length (ECAM 表元素数量不匹配)”？

**原因:** 选择的表格的 ECAM 表索引长度和根据 Gap 计算出的元素数量不同，值得注意的是 ECAM 参数中的 Index start 和 end 需要和 Table 中的元素数量相同。

**解决办法:** 将 Index start 填 1 (均从 1 开始)，End 填写上面括号中计算出的“Table elements”数量。



- **问题 2:** 填写完 ECAM 所有参数后点击“Download To Controller”，为什么会提示“Invalid parameters: ECAMMaster[\*] (ECAMMaster[\*]参数无效)”？

**原因:** 在此界面的 Master definitions 中的 ECAMMaster[\*] 为默认填入的文本“APosRef”，实际上为空，并未真正给 ECAMMaster[\*] 赋值，因此会报告 ECAMMaster[\*] 参数无效，通常只有在用户需要的主数据源为“APosRef”时提示该错误。

**解决办法:** 可以将 ECAMMaster[\*] 先选择为其他除默认的“APosRef”之外的关键字，再将其选回“APosRef”使其为蓝色未生效状态时即可下载参数到控制器。

