教学课件 • 2026.09

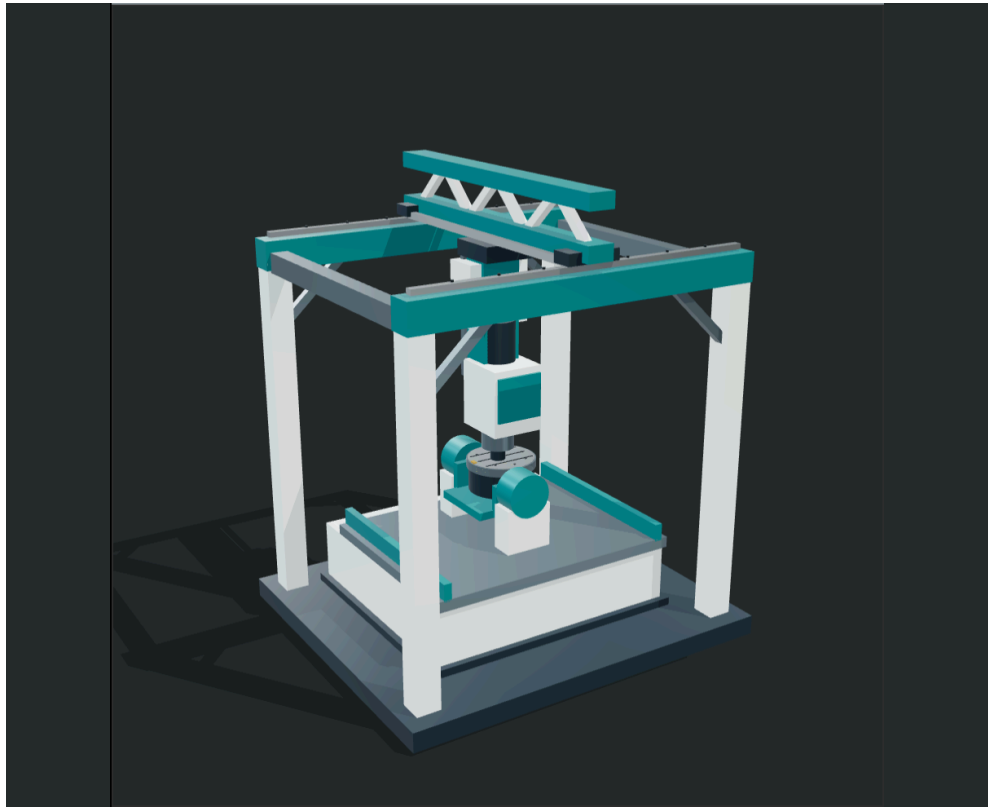
五轴机床与 RTCP

从“各轴独立”到“刀尖跟随”

看懂两个坐标系、一棵结构树、
一条运动学计算链。

适用对象：数控入门、五轴调试与仿真教学
主线机型：XYZAC / XYZBC 双转台桁架机床
建议课时：60–90 分钟，可拆成两讲

原理图解 + 数值演算 + 浏览器实验 + 练习答案



原创 BC 桁架模型：XYZ 移动刀具，B–C 转动工件

> 核心问题：程序中的 XYZ，究竟是在告诉谁“去哪里”？

图：本项目原创 GLB 模型；数值：LinuxCNC WASM 实算。

你的两点理解，怎样说得更准确？

① 非 RTCP：直接轴映射

在本课的双转台结构中，XYZ 带动刀具，A/C 或 B/C 带动工件。

只改变旋转轴指令，不自动触发 XYZ 的刀尖保持补偿。但多个轴仍可在同一段 G 代码中协调插补。

“独立”描述几何映射，不是运动时间完全无关。

② RTCP：保持刀尖相对工件的轨迹

控制器根据编程点、转台角度、旋转中心和标定参数，计算实际关节位置。

工件坐标方向随转台变换，XYZ 关节按需联动。计算还包括平移与偏置，不能只旋转方向箭头。

“跟随”是相对工件满足目标关系，不是世界坐标中的刀尖永远不动。

课程约定：先用零工件偏置、零刀长差的简化模型讲清原理，再加入实际参数。

[S1][S2]；结合本项目 AC/BC 模型。

五轴 = 三个直线轴 + 两个旋转轴

名称	通常的轴线含义	本课中的执行机构
X / Y / Z	沿三个正交方向平移	刀具侧的直线滑台、滑枕
A	绕 X 方向的轴线转动	AC 机床的外层倾转轴
B	绕 Y 方向的轴线转动	BC 机床的外层倾转轴
C	绕 Z 方向的轴线转动	装在倾转台上的内层旋转轴

AC 型： X、Y、Z、A、C

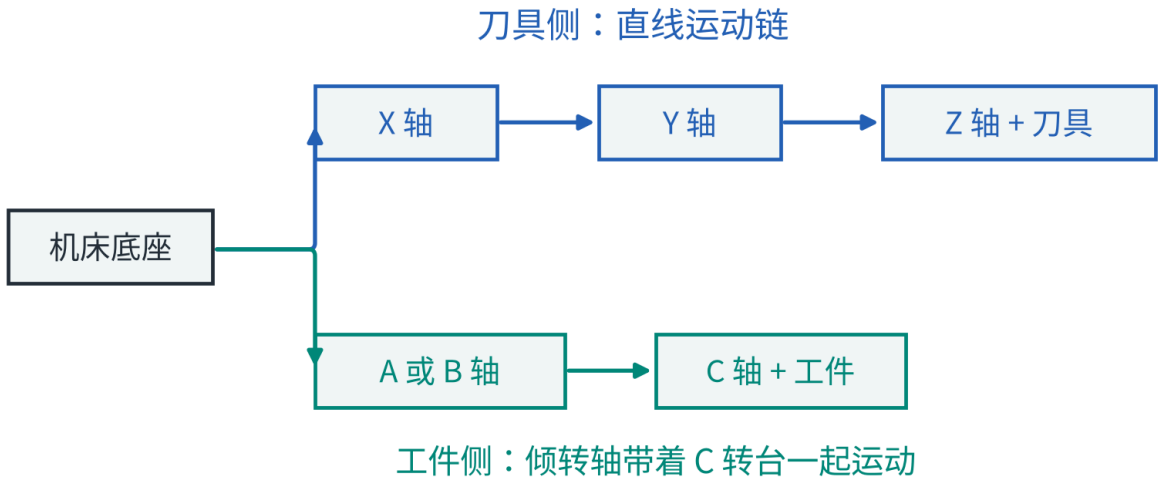
BC 型： X、Y、Z、B、C

XYZABC 六个独立轴同时配置，通常应称六轴。A/B/C 是旋转轴的通用名称，本课不是三只转台轴同时工作。C 的轴线会被父级 A/B 倾转，倾转后不再固定平行于机床 Z。

> 从正方向看懂箭头前，先确认观察视角、父级坐标系和机床正负号约定。

[S3] 轴命名；图中数值采用本项目原生变换约定。

两条运动链，共用一个底座



本课桁架结构：X 带动整个横梁，Y 带动滑座，Z 带动滑枕与刀具。

刀具分支

底座 → X → Y → Z

运动从父节点传给所有子节点。

工件分支

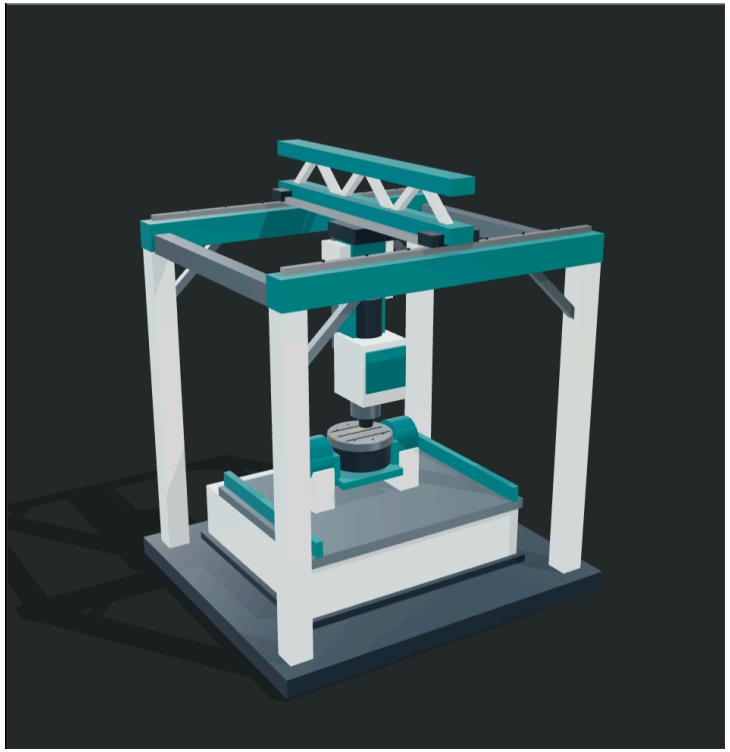
底座 → A/B → C → 工件

A/B 转动时，C 的轴线也随之倾斜。

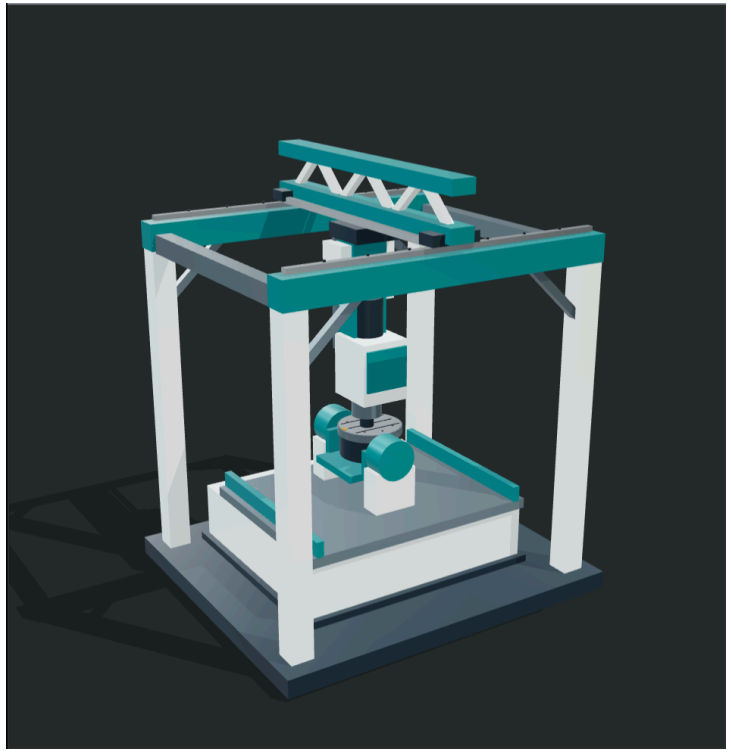
> 这是一种五轴结构，不是所有五轴机床的通用拓扑；摆头、头台混合型需要不同模型。

本项目 trt-xyz-gantry/1；旧标准型在 Z 下另有固定主轴箱节点。

AC 与 BC：外层倾转轴不同



AC 桁架：A 绕 X，内层 C 随 A 倾转



BC 桁架：B 绕 Y，内层 C 随 B 倾转

两者都能实现刀尖相对工件的五轴轨迹，但同一角度数值对应的空间姿态不同；不能只把程序里的 A 字母换成 B，就认为加工姿态完全相同。

> 先识别“哪根轴装在哪根轴上”，再选择运动学模型。

图：原创可下载 GLB；不是制造商的标定尺寸图。

不要把“程序点、关节位置、实际反馈”混为一谈

量	回答的问题	典型表示
编程位置 p	刀尖相对所用编程参考系要到哪里?	(X, Y, Z)，加 A/C 或 B/C
关节指令 q	各执行轴需要到哪个位置?	(J0, J1, J2, J3, J4)
实际反馈 q_actual	编码器报告电机/机械轴在哪里?	真实设备中可能存在跟随误差
三维刀尖位置 r	把关节和模型组合后，刀尖在场景哪里?	显示坐标，不是独立运动控制源

理想仿真中“关节指令 = 关节反馈”常成立，但真实机床两者仍须区分。RTCP 的运动学把 p 转成 q；模型再把 q 投影成画面。

> “实际位置”一词要说明是在说编程坐标、关节指令，还是编码器反馈。

非 RTCP 时，G 代码也不总等于机械绝对位置

程序字值 → 单位 / G90–G91 → 工件零点 / 补偿 → 运动学 → 关节指令

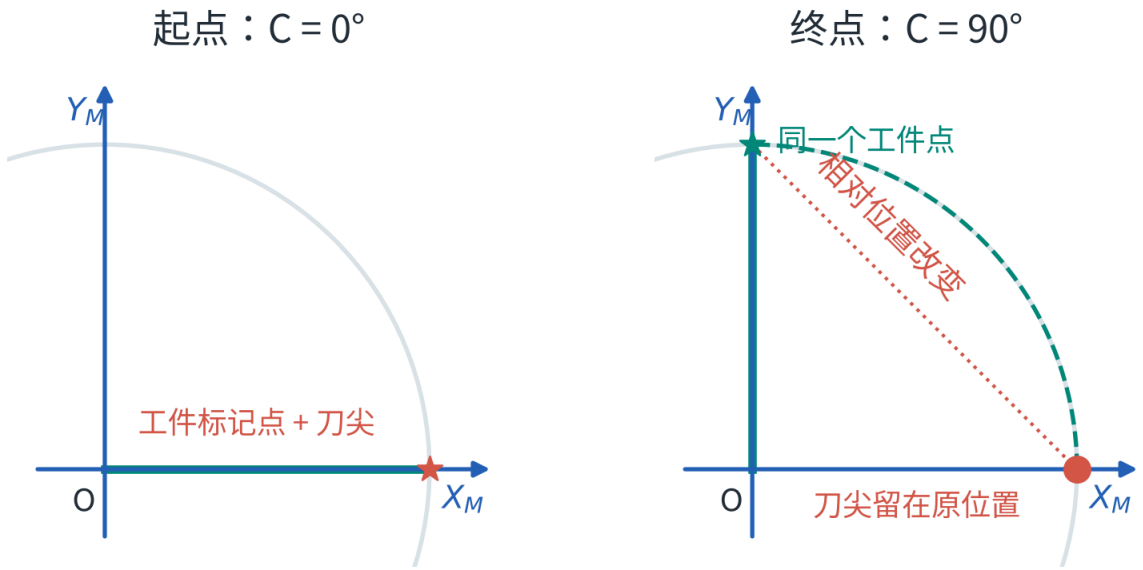
因素	为什么不能直接等同
G90 与 G91	G90 通常给绝对目标；G91 给相对增量，X10 不一定是到 X=10。
G54 等工件坐标系	工件零点可能偏离机床零点；程序 X10 可能对应机床 X30。
G92、刀长等补偿	控制器还需处理偏置和刀具参考点。
Identity 直接映射	在完成以上处理后，所选坐标与关节才按配置直接对应。

简化例：G54 原点位于机床 (20,30,0)，无其他偏置/补偿，程序点 (10,0,5) 对应机床点 (30,30,5)。

> 只有把偏置、单位、增量方式、刀具基准都约定好，才能说“程序值直接对应轴位置”。

[S2][S3]；本页示例为无旋转的 Identity 情形。

只转 C，工件点会离开刀尖

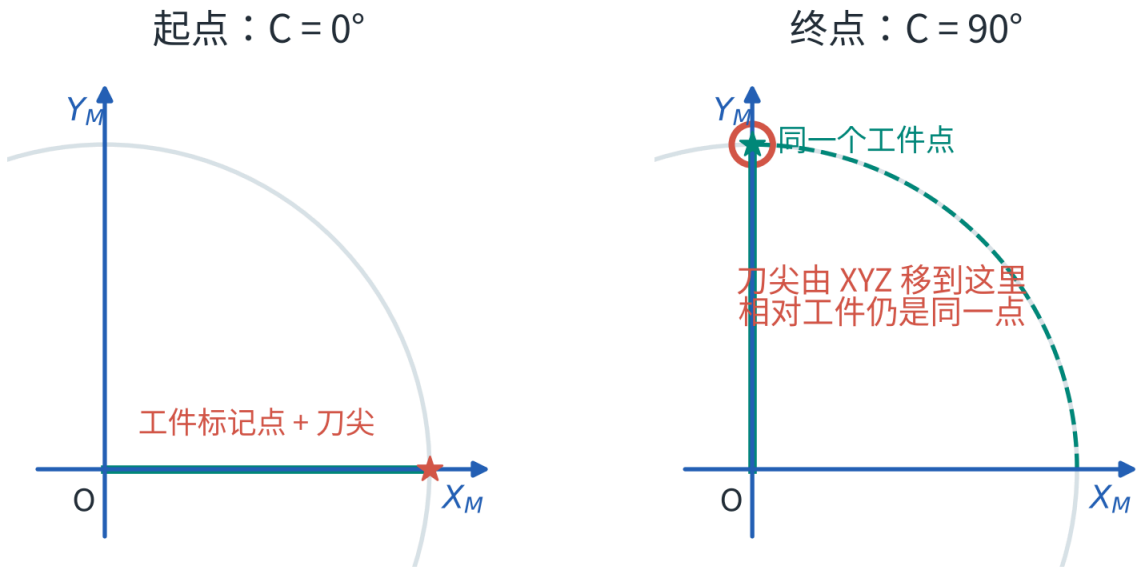


俯视图：蓝色为固定机床坐标；绿色为工件；红色为刀尖。Z 保持 20 mm。

起点刀尖对准转台上半径 100 mm 的标记点。只命令 C 从 0° 转到 90°，**XYZ 关节不补偿**，标记点转到另一处，刀尖仍在原来的机床位置。

> 轴可以独立驱动，不代表“刀具与工件的相对位置”不受其他轴影响。

同样只改变角度，刀尖跟住工件点



同一个工件局部点 p=(100,0,20)，C=0° 与 C=90°；旋转中心和偏置取零。

保持编程点 p 不变，只改变 C。控制器计算所需 XYZ 关节，让刀尖到达标记点的新机床位置。**点在工件坐标中不变，在机床坐标中沿圆弧移动。**

> 判断“跟随”的参考系是工件，而不是屏幕或机床底座。

C 转 90°：同一编程点，不同关节位置

零偏置、零刀长差，p = (100, 0, 20) mm；A/B = 0°。

C 角度	RTCP 编程 XYZ	RTCP 所需关节 XYZ	非 RTCP：只转 C 的关节 XYZ
0°	(100, 0, 20)	(100.000, 0.000, 20.000)	(100, 0, 20)
45°	(100, 0, 20)	(70.711, 70.711, 20.000)	(100, 0, 20)
90°	(100, 0, 20)	(0.000, 100.000, 20.000)	(100, 0, 20)

$$q_x = 100 \cos C \qquad q_y = 100 \sin C \qquad q_z = 20$$

C=90° 时，XYZ 关节目标是 (0,100,20)。这不是控制器偷偷修改了程序点，而是同一点在另一参考系中的正确位置。

> RTCP：程序坐标可以不变，执行轴位置必须按姿态变化。

numeric-evidence.json；原生 WASM IK/FK 双向核对，显示保留3位小数。

C = 45° 时，“沿工件 X 走”意味着什么?

蓝色 X_M、Y_M 是机床固定方向；绿色 X_W、Y_W 随工件转动。

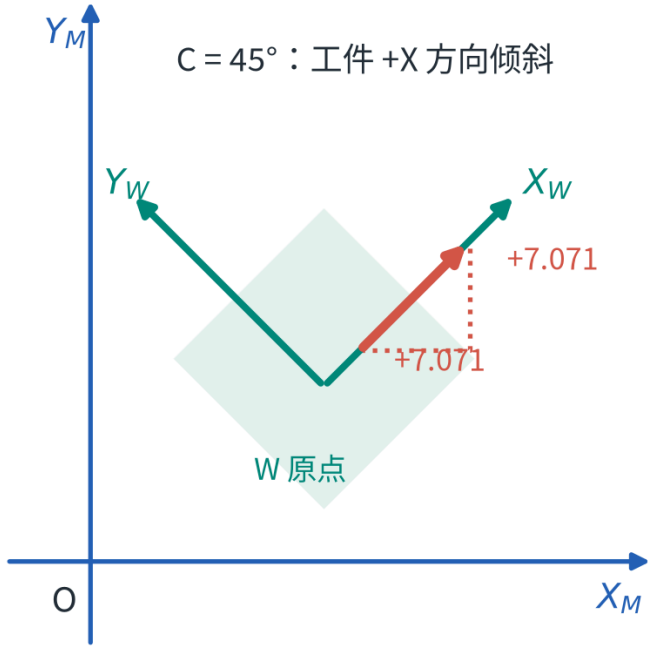
姿态保持 C=45°，编程增量：

$$\Delta p_W = (10, 0, 0)$$

对应的机床直线轴增量：

$$\Delta q = (7.071, 7.071, 0)$$

不是机床 X 导轨真的旋转了，而是 X、Y 导轨合成出所需空间运动方向。



图中工件原点为便于观看作了平移；增量不受原点平移影响。

> 你的“XYZ 方向随旋转改变”可理解为工件基向量在机床坐标中的方向改变。

BC 倾转后，沿工件 X 会同时改变机床 X、Y、Z

采用本课的正向变换约定：

$$R = R_Y(B) \cdot R_Z(C)$$

当 B=30°、C=45° 时：

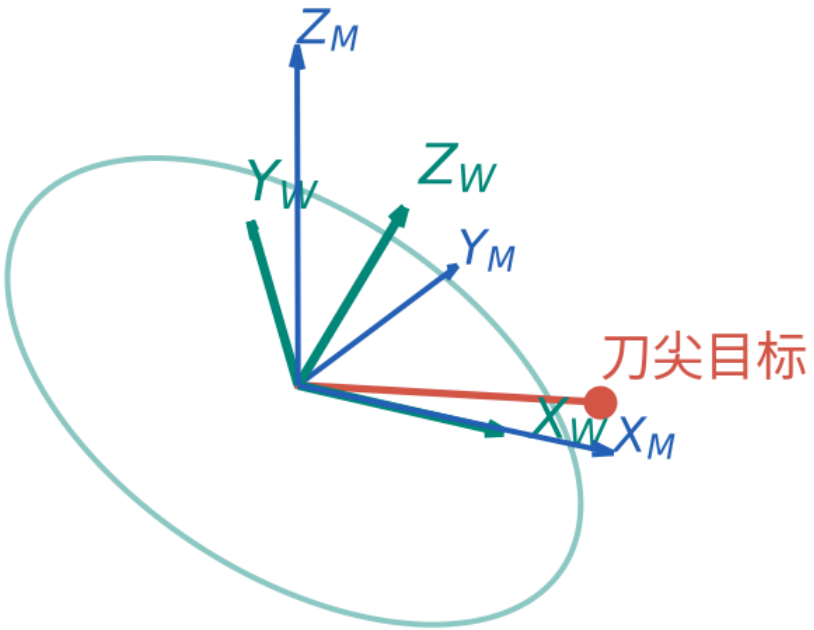
X_W 在 M 中的方向

$(0.6124, 0.7071, -0.3536)$

所以沿 X_W 前进 10 mm：

$$\Delta q \approx (6.124, 7.071, -3.536)\ \text{mm}$$

倾斜后的坐标方向并不与任何一根直线导轨重合，需要三根直线轴合成。



B=30°、C=45°；红点为工件点 p=(100,0,20) 的机床位置。

> RTCP 中“沿工件 X”不是“只让 X 电机动”。

同样的 30° / 45°，不是同一个姿态

仍取 p=(100,0,20) mm，所有旋转中心、偏置和刀长差为零。

结构	父级倾转 × 子级旋转	倾转30°、C45°时的关节XYZ
AC	$R_X(A)$ · $R_Z(C)$	(70.711, 51.237, 52.676)
BC	$R_Y(B)$ · $R_Z(C)$	(71.237, 70.711, -18.035)

AC：绕 X 倾转

C 先改变平面位置，A 再把局部 Y/Z 方向倾斜。

BC：绕 Y 倾转

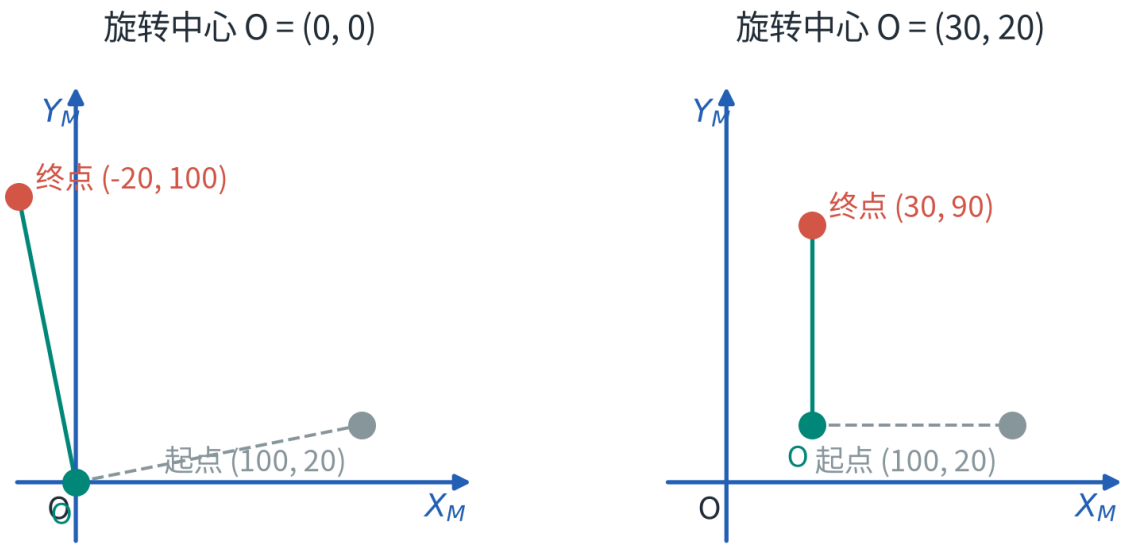
C 先改变平面位置，B 再把局部 X/Z 方向倾斜。

矩阵从右向左作用于列向量；这是坐标转换次序，不是要求 G 代码先运动 C 再运动 A/B。插补时两轴可以同时运动。

> 运动学类型选错，即使每个轴都会动，刀尖也不会沿正确的相对轨迹走。

numeric-evidence.json；零偏置原生 WASM 结果。

同样转 90°，绕不同的中心会到不同的位置



二维俯视例：起点相同，旋转角度相同，仅旋转中心不同。

$$p_M = O + R \cdot (p - O)$$

计算顺序：先减去旋转中心 → 旋转 → 再加回中心。不能只计算 R · p，否则相当于错误地绕坐标零点转。

> RTCP 不只是“方向变换”，还包括“绕哪里转”的平移关系。

C 轴装在倾转轴上：父子关系不能交换

先看几何，再写方程

P：C 轴零位参考点。
D：倾转轴与 C 轴之间的偏置。
T=P+D：倾转轴参考位置。

C 旋转后的点还要随 A/B 倾转；偏置也必须放在正确的父子层级中。

URDF / GLB 里的对应关系

节点	本项目零位 origin
A/B，父节点为底座	P + D
C，父节点为 A/B	−D
work，父节点为 C	−P

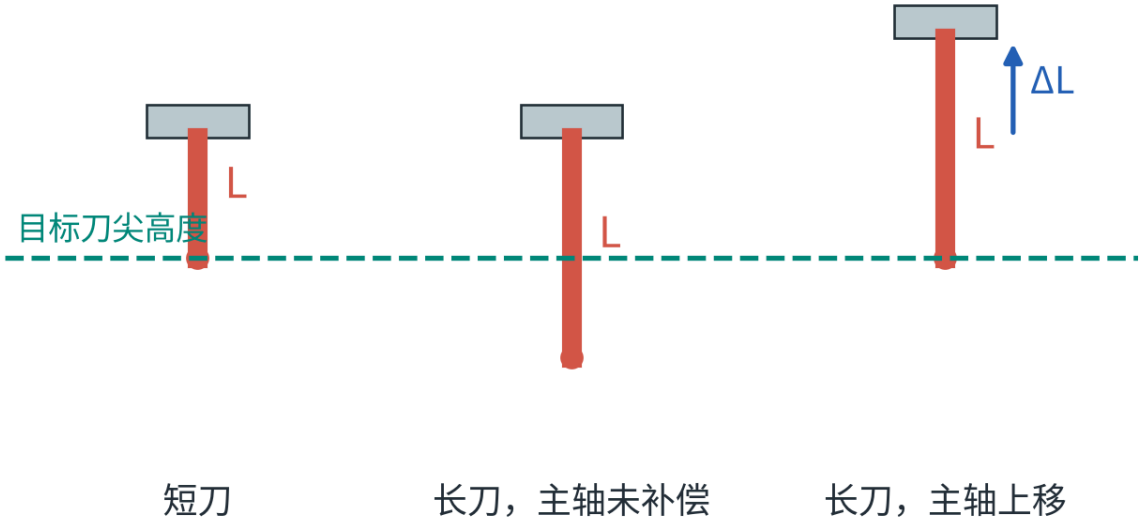
$$q_{XYZ} = P + D + R_{tilt}\,[\,R_C\,(p - P) - D\,] \quad \text{（刀长差为零）}$$

这是本项目 AC/BC TRT 约定。更一般的机床使用完整的刚体变换链；不能把这三个参数关系硬套到摆头机床。长度在运行配置中为 mm，URDF 为 m。

> $R_{tilt}R_C$ 通常不等于 R_CR_{tilt}；机械安装顺序决定计算顺序。

本地 trtfuncs.c；本项目 URDF profile / D11 / D12。

控制的是刀尖，不只是主轴箱



侧视示意：同一主轴参考位置，刀长不同会产生不同刀尖位置。

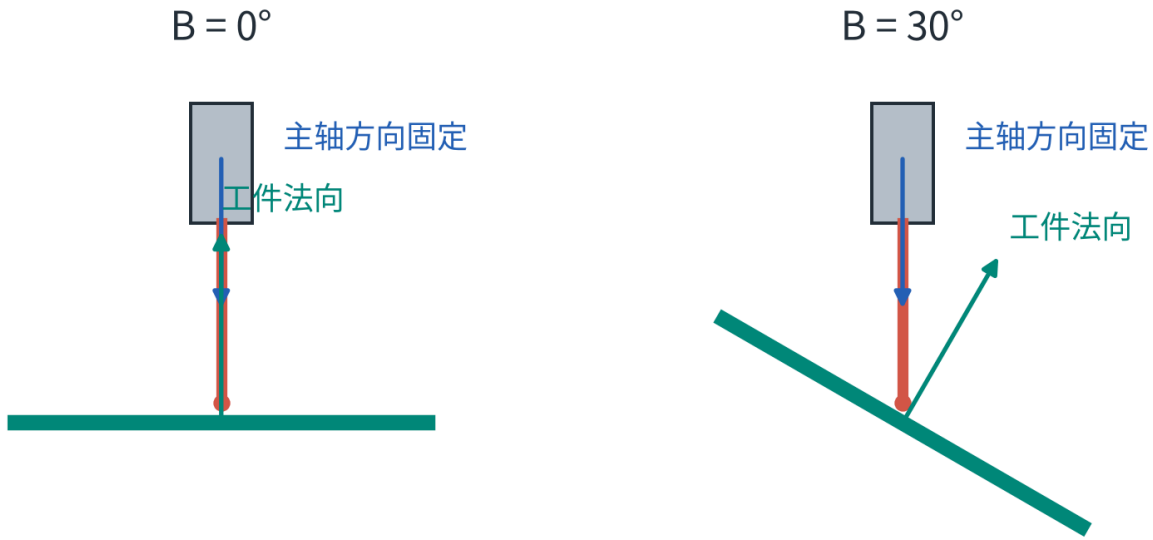
刀长改变时，为保持同一个刀尖目标，需要重新计算主轴参考点的位置。必须明确刀具长度的测量基准、正负号、刀补状态，以及模型上“tool”节点代表哪里。

教学主算例采用刀长差为零。本项目原生 TRT 使用相对于 master tool 的 tool-offset 约定，不能把它直接当作任意模型网格的总长度。

> 同一组关节位置 + 不同刀具标定，不一定得到同一个刀尖位置。

[S1][S3]；实际使用以机床和控制器的刀具标定约定为准。

双转台没有转动主轴，却改变了相对刀轴方向



侧视：主轴在机床中保持竖直，工件随转台倾斜，二者的相对夹角改变。

在本课双转台机床中，刀轴在机床坐标里的方向固定；工件转动后，**刀轴在工件坐标里的方向改变**。RTCP 根据指定角度协调刀尖位置，不会自动推断曲面法向。

> RTCP 不是“自动让刀轴垂直于任意曲面”；姿态仍由程序、CAM 或操作者给定。

分清“保持点”“沿方向走”“同时改变位置和姿态”

输入变化	RTCP 应做什么	观察重点
XYZ 不变；A/B、C 改变	让刀尖保持同一工件编程点；XYZ关节可能联动。	刀尖与工件标记点不分离。
角度不变；XYZ 改变	沿当前工件参考系定义的方向移动。	机床多轴合成编程方向。
XYZ 与角度同时改变	执行变化的编程轨迹，同时维持角度对应的几何关系。	全过程相对轨迹正确，不只看终点。
XYZ 不变；点恰在 C 轴线上；只转 C	该点机床位置可以不变。	XYZ 不动不一定是故障。

“五轴联动”不要求每个时刻五轴速度都非零。非 RTCP 也能多轴同步加工：CAM/后置可以预先按指定机床几何计算轴目标，但这不等于控制器开启了实时刀尖补偿。

> 每次解释一个动作，先说清哪些程序量保持不变、哪些程序量在改变。

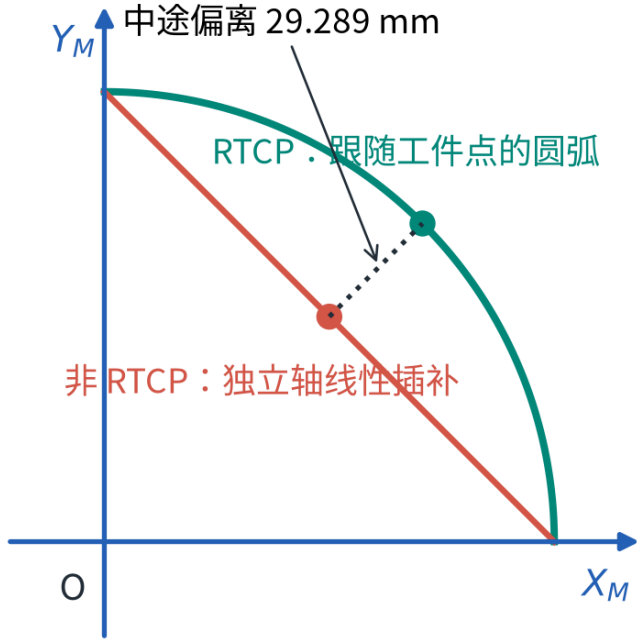
终点都正确，不代表中间轨迹正确

同一个工件点绕 C 转动，理想半径为 100 mm。

RTCP：刀尖跟随圆弧。
不做 RTCP：若只让 X/Y 线性插补到相同终点，刀尖可能走弦线。

中间位置	X / Y
圆弧，C=45°	70.711 / 70.711
弦线，50%进度	50.000 / 50.000

图示采用共同线性进度比较几何路径，不是断言所有控制器的加减速过程相同。



中途差距 29.289 mm；起点和终点却完全相同。

> 验收 RTCP 要看连续采样、刀尖相对轨迹和关节约束，不能只比首末坐标。

M429 ↔ M428：坐标可以变，机床不能无故跳动

本项目 BC 示例：保持同一组物理关节，切换坐标解释。

量	M429 / Identity	M428 / TCP
物理关节 XYZ	(0.000, 0.000, 44.586)	(0.000, 0.000, 44.586)
物理关节 B / C	4.8675° / 179.88°	4.8675° / 179.88°
显示/解释的 XYZ	(0.000, 0.000, 44.586)	(-29.885, 27.975, 43.985)
三维刀尖	同一空间位置	同一空间位置

取当前关节 q → 切换运动学 → 新模式正解 p → 同步 TP / 解释器

本例使用真实 BC 模型参数 P=(−18,14,12)、D=(22,0,−40) mm，刀长差0。切换之后再给运动指令，才产生运动。

> 若切换时强行保持旧 XYZ 数字并重新逆解，就可能错误地改变物理关节。

[S2] 同步原则；本地 control.c::handle_kinematicsSwitch；WASM实算。

RTCP、旋转工作平面、刀补，不是同一个开关

功能	主要解决的问题	不要误认为
RTCP / TCP 运动学	姿态变化时，按机床结构实现目标刀尖相对轨迹。	所有控制器都会以完全相同方式解释XYZ。
工件坐标系 / 倾斜工作平面	选择编程参考原点或定义编程方向。	只旋转工作平面就自动具备完整五轴RTCP。
刀长补偿	把刀具参考点换算到实际受控刀尖。	更换刀具后只更新网格外观就够了。
机床标定	确定中心、偏置、轴向、正负号和长度。	只要支持RTCP，错误几何参数也会自动纠正。

不同控制器可能分别使用 TCP、TCPM、RTCP 等名称，并把倾斜平面/刀补组合进不同指令。本课明确讲本项目的 AC/BC 双转台语义，不推定所有品牌一致。

> 换控制器时先核对“参考系、受控点、角度定义、指令组合”，不要只对照功能名称。

同样写 C90，开关状态决定 XYZ 是否补偿

M429：直接映射

```
G21 G90 G94
M429
G1 X100 Y0 Z20 A0 C0 F300
G1 C90 F60
```

后一段只改变 C。
XYZ 关节保持 (100,0,20)，
工件标记点会离开刀尖。

M428：RTCP

```
G21 G90 G94
M428
G1 X100 Y0 Z20 A0 C0 F300
G1 C90 F60
```

后一段保持编程 XYZ。
XYZ 关节变到 (0,100,20)，
刀尖仍对准同一个工件点。

仅为零偏置/零刀长差 AC 教学片段，起始角度和当前模态需已知；BC 用 B0。现有下载模型参数非零，不能期待相同数字。M428/M429 为本项目重映射，不是通用标准M码；F值只作插补示意，实际需按机床的旋转/混合进给规则设定。

> 程序字值相同，不代表开关前后表达的是同一个机械目标。

本项目 xyzac/xyzbc-trt.ini，428remap.ngc / 429remap.ngc。

用 AC/BC 桁架模型验证，而不是只看公式

1. **加载：**打开“导入机床模型”，选 AC 或 BC 桁架示例；记录旋转中心与偏置。
2. **准备：**确认空闲，开机并回零；选一个离 C 轴线有一定距离的刀尖位置。
3. **Identity：**选择 M429，只 Jog A/B 或 C；观察 XYZ 关节是否保持。
4. **切换：**停止后选择 M428；观察关节/刀尖不跳，XYZ 坐标可能变化。
5. **TCP：**再次 Jog 旋转轴；观察 XYZ 关节按需补偿、刀尖相对工件点保持。
6. **对照：**先停止，再切回 M429；重复方向相同的小幅 Jog，比较两种行为。

实验先用小幅角度并远离限位。记录的是“每次试验起点相对工件的同一点”；切换或重新定位后，不要把两个不同起点混作同一个目标。

> 同时观察三项：程序XYZ、J0…J4、刀尖与工件的相对关系。

改几何参数后，为什么仍应保留同一个模型？

正确的数据流

参数表：中心 / 偏置

↓

有效 URDF：节点零位关系

↓

INI → Worker → WASM

↓

关节 → 原导入模型

课堂观察

- 同一 GLB 的网格、材质和文件身份保持。
- 新的轴心与偏置应进入运行结构和运动学。
- 相同程序点和角度，可能得到不同的 XYZ 关节值。
- 参数改变后需重新回零并重新解析程序。

源GLB不自动改写；可保存INI记录参数。改变轴心不等于自动重建支架、铸件等CAD外形，也不提供碰撞检测。

> 模型保留和参数生效必须同时成立；只改外观或只改计算都会造成误导。

本项目 D11 / E09：AC、BC各18个姿态/刀长样本验证。

“工件点可达”不等于“每个关节都没到限位”

两层约束

约束	检查什么
坐标限位	编程轴/坐标允许的范围
关节限位	实际执行机构的物理行程
动力学约束	速度、加速度等能力

RTCP 旋转时，即使编程 XYZ 不变，补偿出来的 X/Y/Z 关节仍可能触及边界。

本项目排障实例

BC 配置允许较大的 B 坐标范围，但对应 JOINT_3 上限为 **50°**。

旧问题：越界 → 运动计算失败 → 事件时钟不同步 → Connection ERROR。

修复后：按关节约束停止，保留正确时钟，可松开并反向 Jog。

本仿真对 RTCP 越界采用最后有效伺服样本停止，不等同于工业控制器的全路径限位预测与减速保护。范围和行为必须分别验证。

> 正常限位阻止是一条运动诊断；序号缺口或连接错误是软件故障，不是RTCP原理。

六个问题，快速判断是否真正理解

说法	准确解释
“非RTCP时各轴互相没有影响。”	没有自动补偿；但工件转动当然会改变它与刀具的相对位置。
“RTCP把X/Y/Z导轨转了。”	导轨不转，是工件基方向改变，由固定导轨合成空间运动。
“RTCP时XYZ永远不变。”	只在保持同一编程点时程序XYZ不变；关节XYZ通常变化。
“只要开RTCP，刀尖在空间就不动。”	刀尖可在机床空间移动，以保持它相对工件的目标关系。
“C转了但XYZ没动，一定坏了。”	目标点恰在C轴线上时可能正确，应先检查几何。
“切换模式时XYZ数字跳，必须修好。”	要检查关节和物理刀尖是否跳；新参考系里的数值改变可能正确。

> 先说参考系，再说位置；先看结构，再说联动。

请先画箭头，再写结论

1. **点追踪：**零偏置，C=0°，工件点 p=(80,0,10)。只转 C 到90°，分别写出 Identity 和 RTCP 的 XYZ 关节目标。
2. **方向合成：**C=30°且倾转角为0°，沿工件 +X 方向走20mm，机床 X/Y 各应增加多少?
3. **模式切换：**M429切到M428时，XYZ显示值变化、J0…J4不变、刀尖不动，是否正常？为什么?
4. **偏置识别：**BC模型 C 的局部 origin=(-0.022,0,0.040)m，倾转轴 origin=(0.004,0.014,-0.028)m。求本项目 P、D，单位mm。
5. **结构判断：**C节点错误地挂在Z滑枕下。只Jog X时会发生什么？为什么不能靠RTCP开关解决?

> 建议分组讨论 8–10 分钟；答案在下一页。

用“相对谁”检查每一步

1. **Identity**：(80,0,10)；**RTCP**：(0,80,10)。前者刀尖留在机床原位置，后者继续对准同一工件点。
2. **ΔX=20cos30°=17.321mm；ΔY=20sin30°=10.000mm。**方向向量长度仍为20mm，机床Z不变。
3. **正常。**运动学切换重新解释当前物理状态。坐标可以变，关节和刀尖应保持。
4. **P=T+U=(-18,14,12)mm；D=−U=(22,0,−40)mm。**先统一单位，再按父子关系相加，不能直接拿URDF的米当毫米。
5. **C及工件会错误地随XYZ移动。**这是结构树错误，不是开关模式能修复的；需把A/B→C分支正确挂回底座。

检查顺序：轴数 → 层级 → 单位/正负号 → 几何参数 → 运动学 → 程序与状态。

从一根 C 轴，推到两根旋转轴

绕 C 的平面变换

$$x' = x\cos C - y\sin C$$
$$y' = x\sin C + y\cos C$$
$$z' = z$$

AC：再绕 X 倾转 A

$$q_x = x'$$
$$q_y = y'\cos A - z'\sin A$$
$$q_z = y'\sin A + z'\cos A$$

BC：再绕 Y 倾转 B

$$q_x = x'\cos B + z'\sin B$$
$$q_y = y'$$
$$q_z = -x'\sin B + z'\cos B$$

这里所有旋转中心、轴间偏置及刀长差为零，列向量采用右手正向变换。

式中的 q 是关节目标；若导入的是其他结构、符号约定或参考点，必须重新建立变换链。

先学会解释“这一步在谁的坐标系中发生”，再记忆公式。

与本地 trtfuncs.c 零偏置逆解一致；三角函数实际以弧度计算。

加入旋转中心、偏置与 master-tool 刀长差

$$D_L = D + L\, e_z$$

$$q_{XYZ} = P + D_L + R_{tilt}\,[\, R_C\,(p - P) - D_L\,]$$

符号	AC	BC
R_{tilt}	$R_X(A)$	$R_Y(B)$
D	$(0, D_y, D_z)$	$(D_x, 0, D_z)$
旋转关节	$q_A=A$，$q_C=C$	$q_B=B$，$q_C=C$

P为原生rot-point；L为本项目/原生TRT的tool-offset标定量；$e_z=(0,0,1)$。本式返回关节XYZ，不是未经处理的网格刀尖世界坐标。G54等解释器偏置在输入p形成前已按控制器规则处理。

这一刀长表达是本项目约定，不能替代任意控制器的通用刀补说明。主课例子L=0；实际机床应以标定模型及其完整前后向验证为准。

> 工程实现不在画面中“补一个偏移”；应让同一组结构数据贯穿解释器、运动学和模型。

原生源码：linuxcnc/src/emc/kinematics/trtfuncs.c，xyzac/xyzbcKinematicsInverse。

怎样讲、怎样验证、怎样继续学习

建议授课节奏

10分钟：结构、轴名与三种位置。
20分钟：C轴点跟踪、方向合成、AC/BC倾转。
15分钟：旋转中心、偏置、刀长与模式切换。
15分钟：浏览器实验与错误诊断。
10分钟：练习、答案和模型参数核对。

本课验算

12个姿态由真实WASM逆解计算，并通过正解回代；另核对C45°方向增量、BC切换示例及弧弦差。详见同目录 `numeric-evidence.json`。

图形为原创教学示意及本项目GLB截图，不用于制造尺寸标定。

官方资料与本地实现

[S1] LinuxCNC · 5-Axis Kinematics
AC/BC结构、运动学及偏置。

[S2] LinuxCNC · Switchable Kinematics
模式切换、identityfirst、同步要求。

[S3] LinuxCNC · CNC Machine Overview
轴名、坐标、受控点与协调运动。

本地：trtfuncs.c；control.c；428remap.ngc / 429remap.ngc；docs/glb-machine/。M428=TCP、M429=Identity仅适用于本项目配置。

资料核对日期：2026-09-18。不同版本/品牌的指令与坐标约定应另行核查。

> 一句话记忆：程序描述相对工件的目标，运动学把目标换算成各关节应该怎样动。