

文档密级：非保密

适用范围：对外公开

Tracer 钢结构自动化焊接系统手册

西安知象光电科技有限公司

修订记录

版本	日期	版本描述	备注
V1.0	2021.9.28	版本创建	
V1.1	2021.11.5	内容优化	拍照方向部分
V1.2	2021.12.6	修改内容	新增配置项

前言

手册内容

本手册介绍了 Tracer 钢构焊接系统手册的内容及使用说明。内容包括系统的运行流程、Tekla 插件介绍及使用说明，TracerStudio 使用说明。

使用声明

必须仔细阅读本手册，只有理解和阅读相关部分后，才能更熟练的使用本系统。

联系方式

咨询热线：400-168-1992 （工作日 09:00-18:30）

官方网站：<http://www.chishine3d.com/>

官方微信：



目录

1. 系统介绍.....	6
1.1 运行流程	6
1.2 名词定义	8
2. TracerAssistant.....	9
2.1 插件介绍	9
2.2 插件安装	9
2.3 TEKLA 设置.....	10
2.3.1 设置视图.....	10
2.3.2 设置主零件	12
2.4 焊缝编辑	13
2.4.1 参数设置	13
2.4.2 自动生成焊缝	14
2.4.3 手动创建焊缝	14
2.5 焊缝组编辑.....	15
2.5.1 增加、删除、修改	15
2.5.2 调整工序分配	15
2.6 工序编辑	16
2.6.1 设置工件坐标系.....	16
2.6.2 设置工序坐标系.....	16
2.6.3 设置地面法线	17
2.6.4 焊缝组排序	17
2.6.5 增加、删除、修改	18
2.6.6 生成文件	18
3. TracerStudio 程序编辑	20
3.1 准备工作	20
3.2 功能介绍	21
3.2.1 数据结构.....	21

3.2.2 执行流程	22
3.3 参数配置	23
3.3.1 配置工作站属性	23
3.3.2 设置拍照位参数	26
3.3.3 修改焊接轨迹默认参数	27
3.4 操作流程	28
3.4.1 焊缝信息导入	28
3.4.2 执行计算	28
3.4.3 焊接单元编辑	28
3.4.4 程序导出	29
3.4.5 工程保存	29
3.5 程序模板修改	30
4. 机器人控制	31
附录 A 机器人程序说明	32
A.1 FANUC	32
A.2 Kawasaki	34
A.3 STEP	36

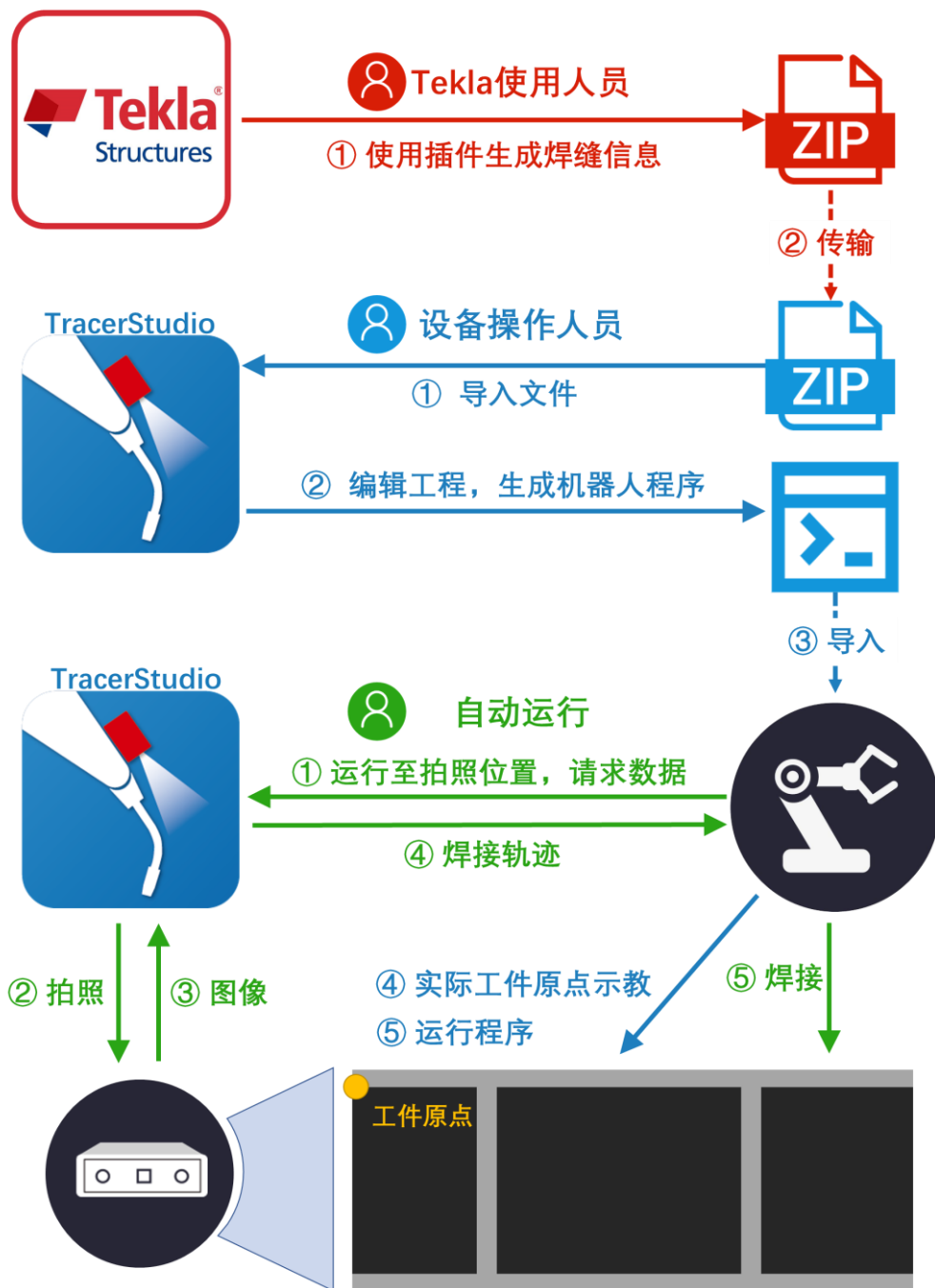
1. 系统介绍

Tracer 钢构焊接系统是由 Takle 插件 TracerAssistant 以及 Tracer3D 焊接视觉系统组成。

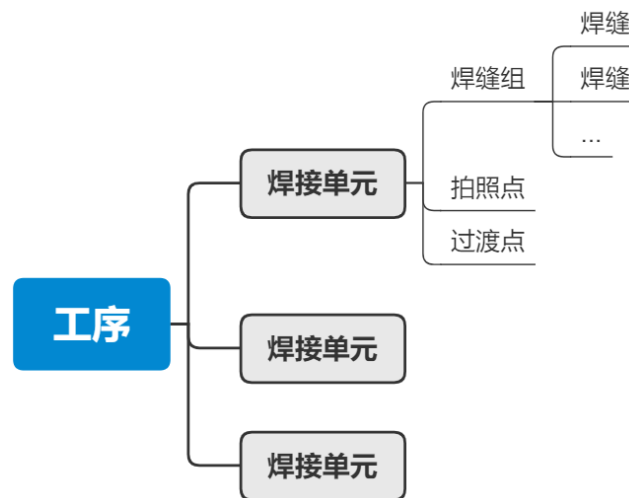
可以完成机器人程序自动离线编程，实现机器人钢构焊接免编程、免示教的焊接自动化。

1.1 运行流程

系统运行流程如下图所示。



1.2 名词定义



参数名称	说明
工序	一个装夹位置所完成的焊接作业为一个工序
焊接单元	1-N 次拍照，1 次焊接为一个焊接单元，包含 2 个过渡点、N 个拍照点
焊缝组	一个焊接单元中包含的焊缝为一个焊缝组
焊缝	板与板之间需要焊接的线段部分
拍照点	相机拍照时机器人到达的点位
过渡点	焊接单元工作前和工作后的运动过渡点

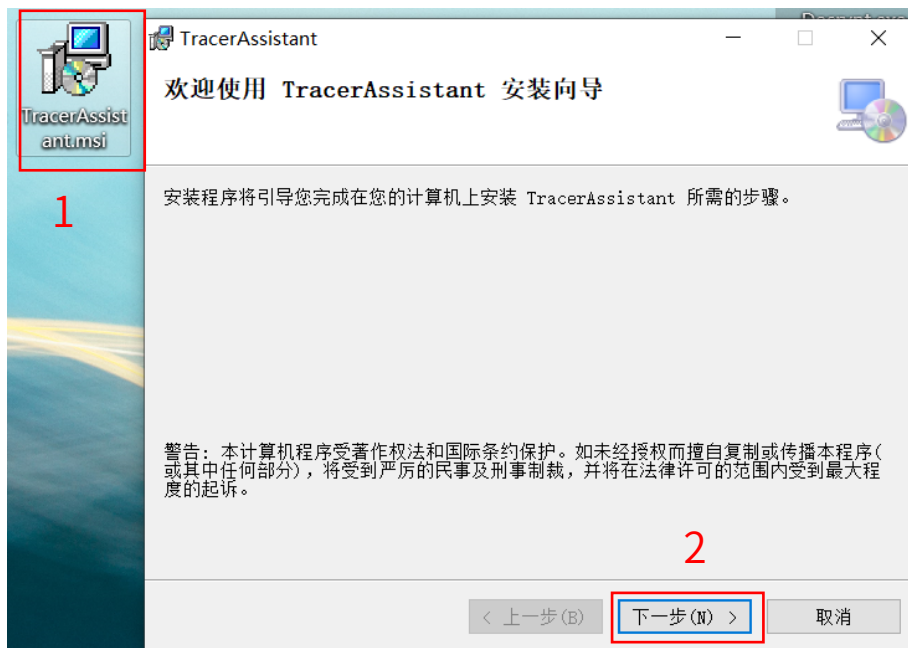
2. TracerAssistant

2.1 插件介绍

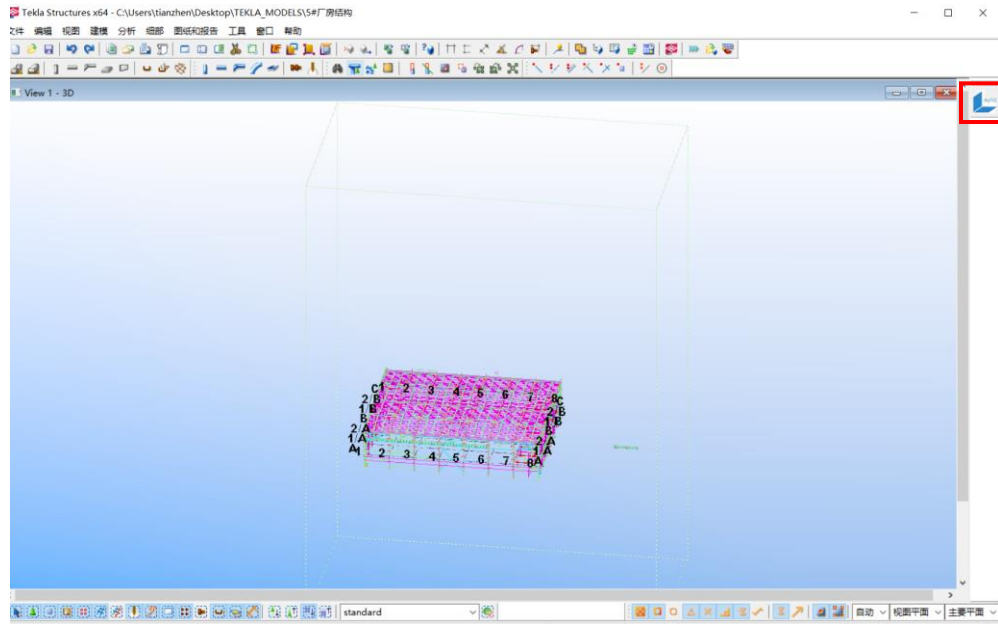
TracerAssitant 插件可以根据工件模型生成焊缝，并对焊缝进行分类、分组、排序，将焊缝数据导出。焊缝数据可用于 TracerStudio 的拍照位计算。该插件的特点是：自动识别、操作简便、自主定制、高效输出。

2.2 插件安装

- STEP1: 打开插件安装包，点击”下一步”完成插件安装。



- STEP2: 打开软件，插件则自动安装在软件右侧。



2.3 TEKLA 设置

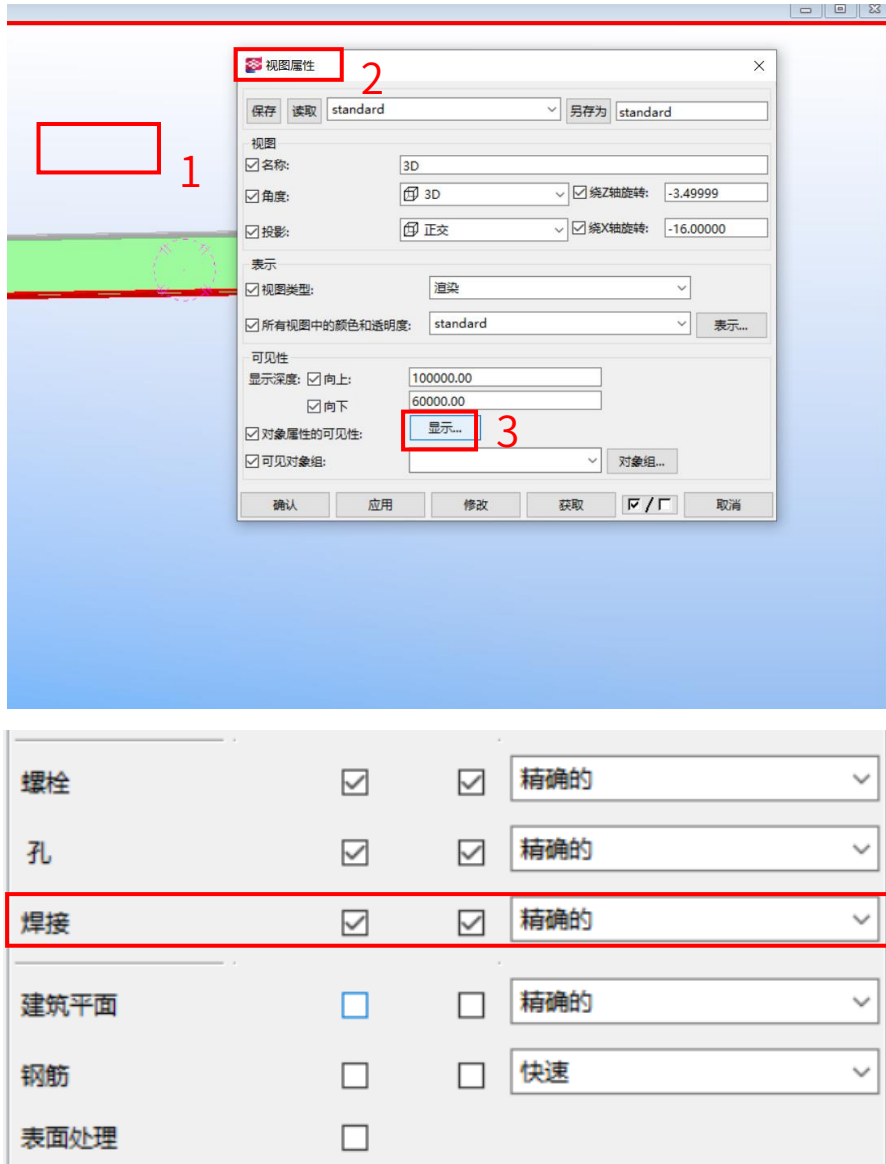
2.3.1 设置视图

- STEP1: 选中视图

双击空白处，外边变红色，代表视图被选中。

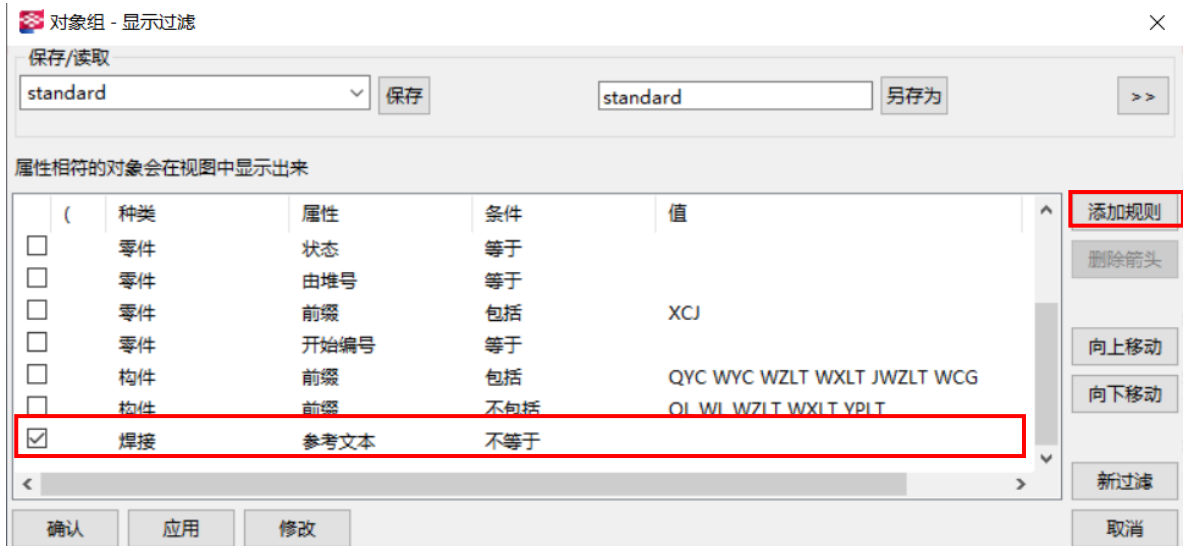
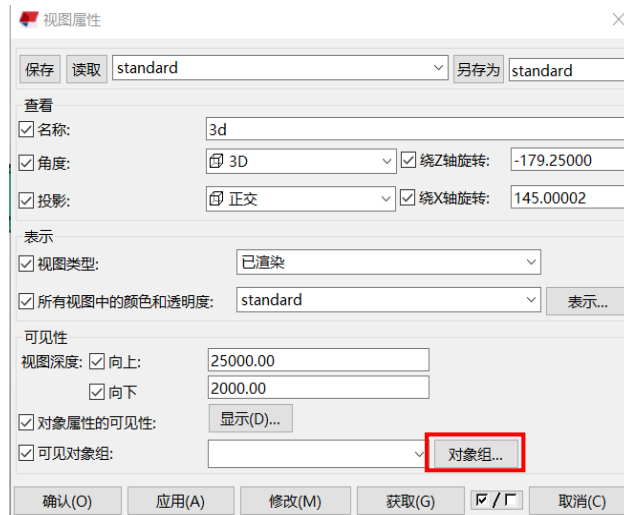
- STEP2: 焊缝显示

点击【显示】按钮，按照图示内容设置焊缝视图参数。

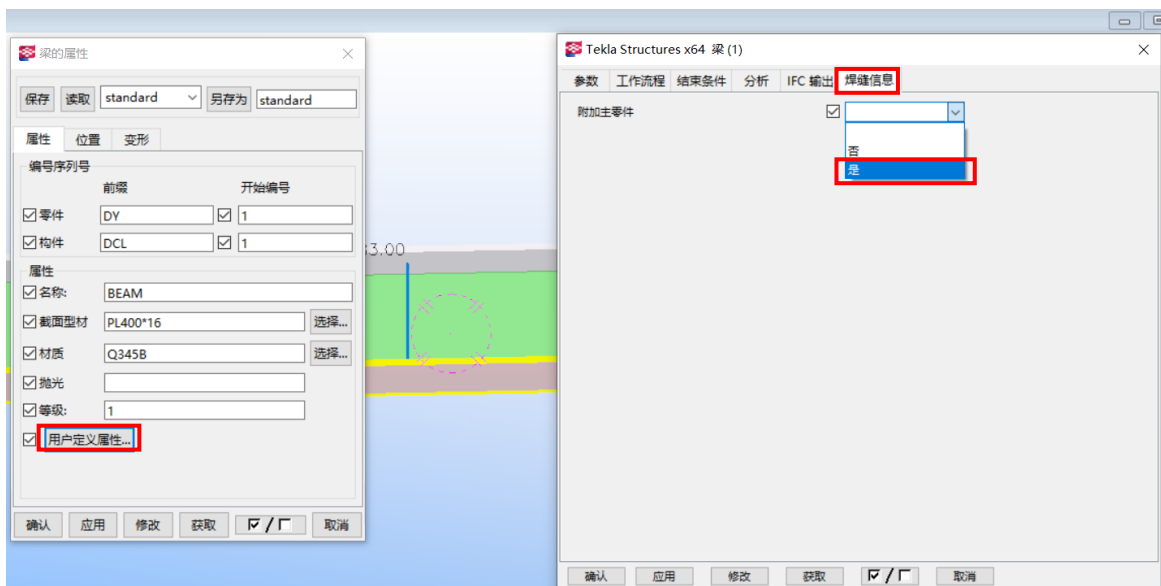


● STEP3: 焊缝过滤

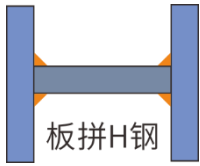

点击对象组按钮，按照图示内容添加并设置焊缝过滤条件。



2.3.2 设置主零件





双击模型上的主零件（钢构的主体零件：H 钢梁、腹板、翼板等），选择用户定义属性，设置焊缝信息，选择主零件“是”。存在多个主零件的情况下，需要逐个设置。

构建类型对主零件定义的影响：			
构件类型		板拼钢构	需要定义至少三个主零件
		H 型钢	需要定义至少一个主零件

备注：双击主零件时，选择构件按钮不能被选中。

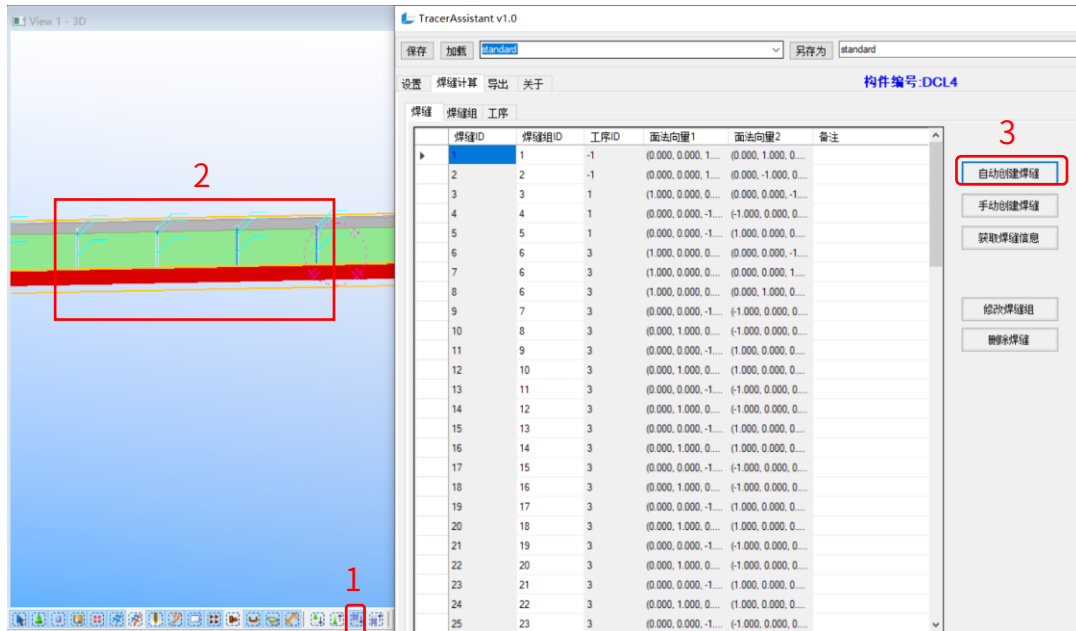
2.4 焊缝编辑

2.4.1 参数设置

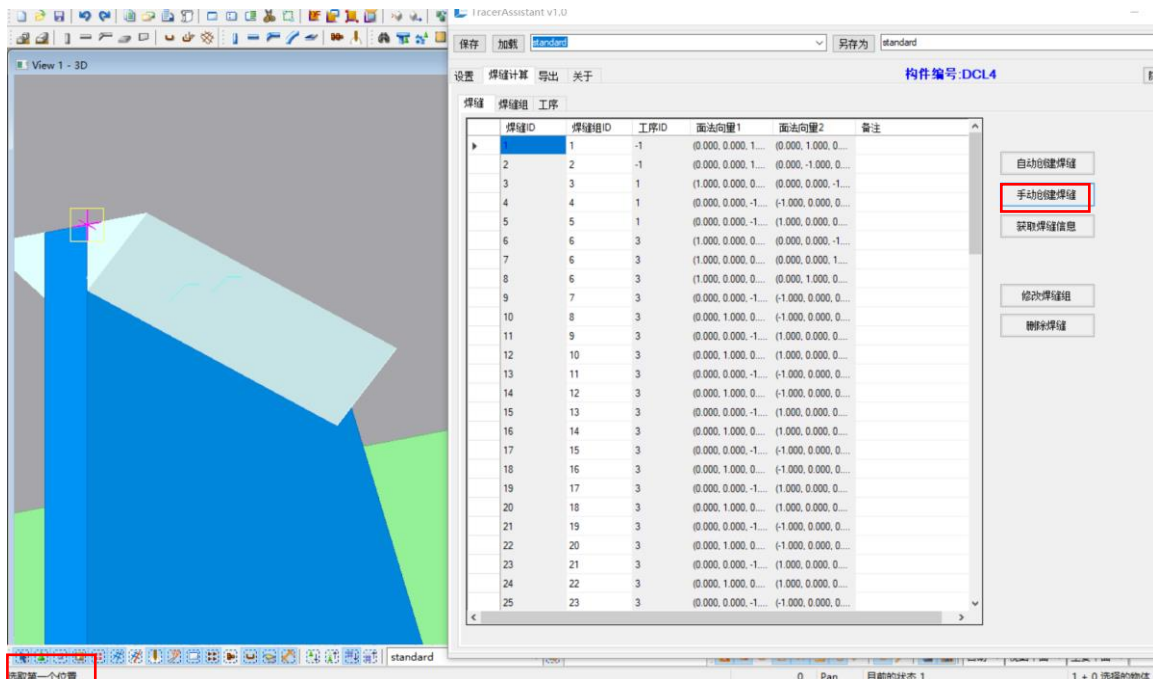
参数名称	说明
焊缝名称前缀	/
焊缝创建间隙阈值	板板之间小于阈值时创建焊缝，大于阈值则不创建
焊缝分组距离阈值	焊缝自动分组时，距离小于阈值的焊缝分为一组
坐标系创建方式	先 Y 后 X \ 先 X 后 Y
构件类型	一体 H 型钢 \ 板拼 H 型钢
焊缝组排序方式	对焊缝组先按照第一顺序进行排序，小于距离阈值的焊缝组序列相同； 对序列相同的焊缝组按照第二顺序进行排序，小于距离阈值的焊缝组序列相同； 对序列相同的焊缝组按照第三顺序进行排序
排序考虑焊接方向	<p>不勾选：</p>  <p>勾选：</p> 

2.4.2 自动生成焊缝

选中选择构件按钮，再选择工件。点击【自动创建焊缝】，生成所有焊缝。



2.4.3 手动创建焊缝



根据提示进行手动焊缝创建。

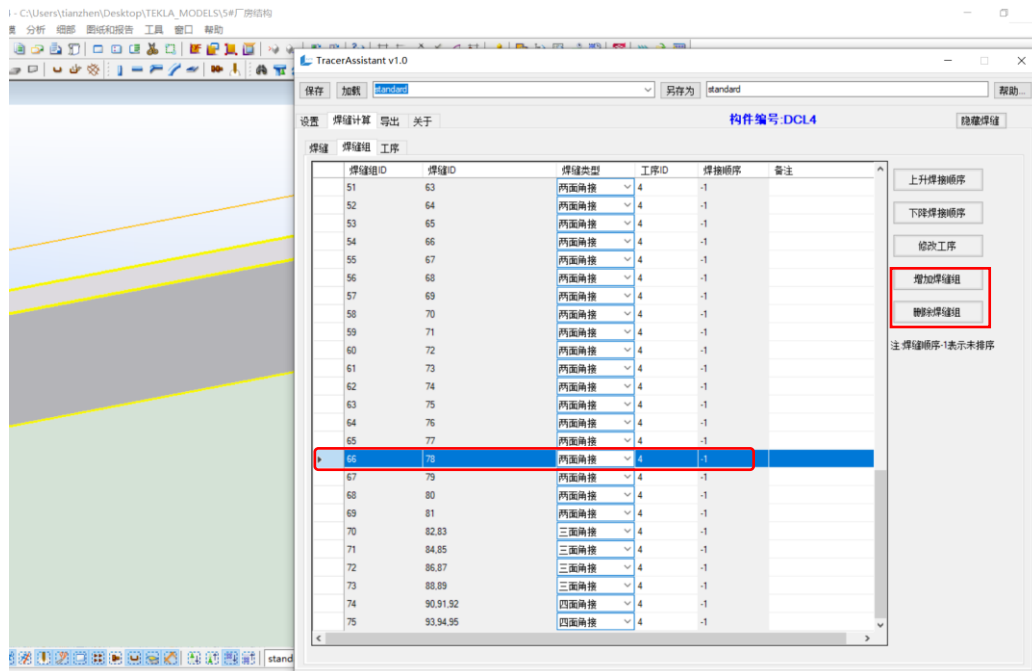
创建过程为：选第一个面，选第二个面，选焊线起点，选焊线终点。创建焊缝成功。

2.5 焊缝组编辑

2.5.1 增加、删除、修改

根据实际加工的工序安排，对焊缝组进行增加、删除和修改。

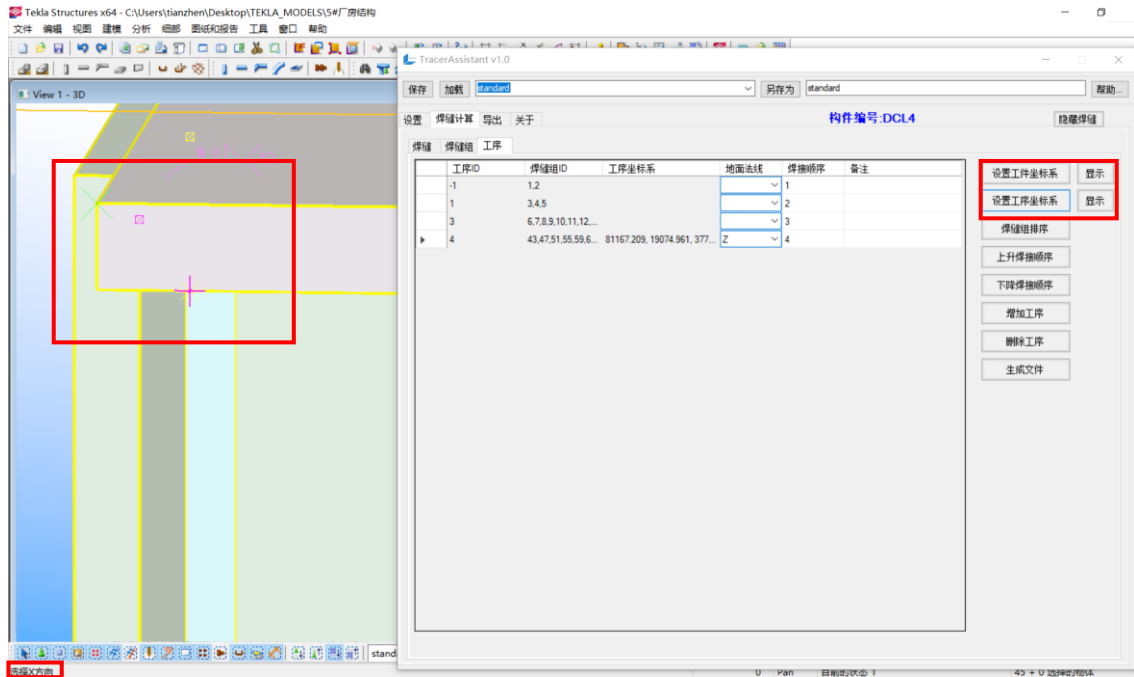
焊缝组的定义是将一个焊接单元的焊缝划归为一组，在拍照后完成焊接。焊缝类型指的是构建焊缝的结构类型。



2.5.2 调整工序分配

合理规划，调整焊缝组的所属工序。

- 按照参数设置的排序规则，对所选工序进行自动排序。
- 使用【上升焊接顺序】【下降焊接顺序】。对焊缝组顺序进行焊接顺序调整。
- 使用【修改工序】。对焊缝组的工序号进行修改。

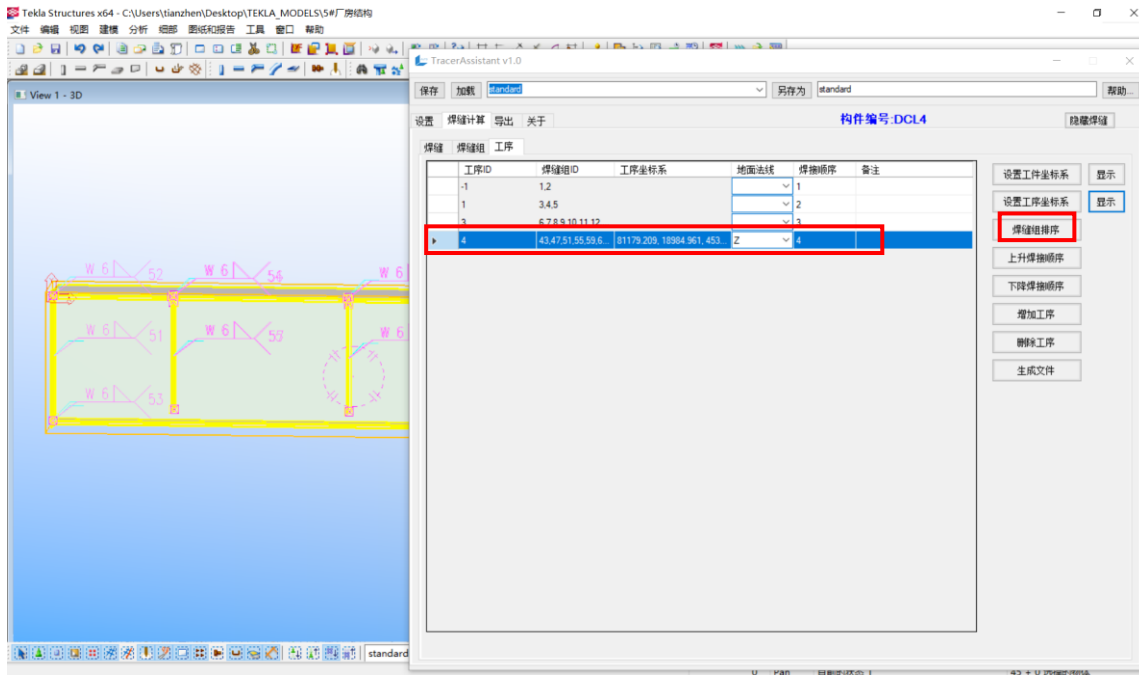


2.6.3 设置地面法线

地面法线，指的是实际工作站的地面法线在工序坐标系中的方向，通俗的讲，是朝上的方向是哪个。

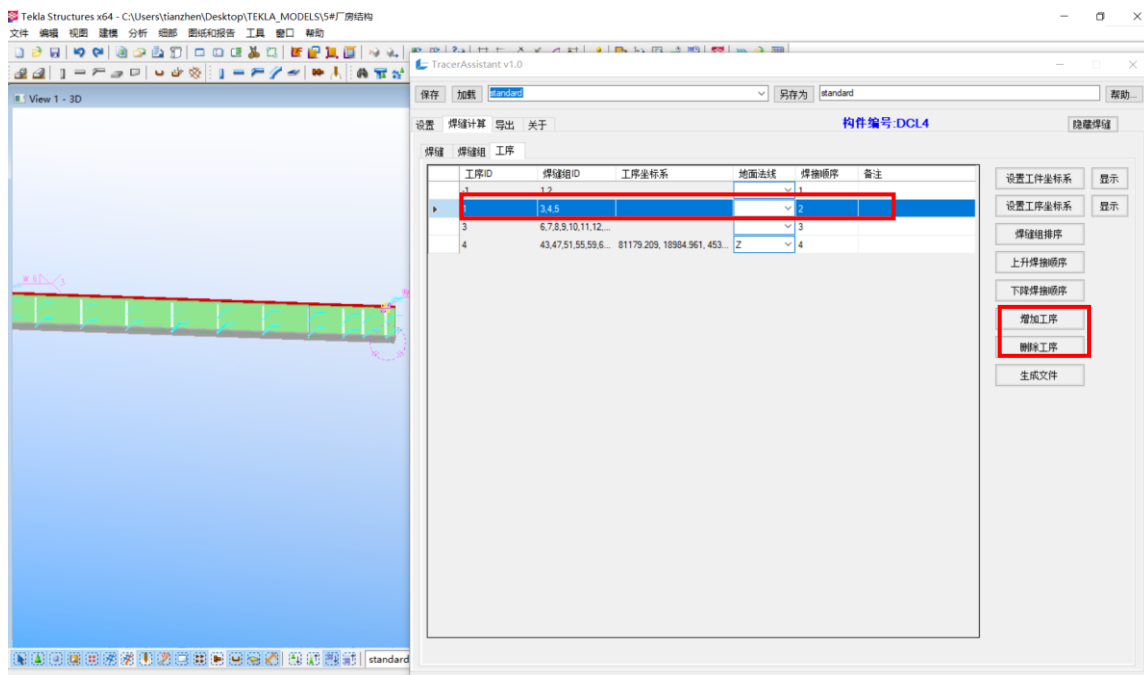
2.6.4 焊缝组排序

根据设置内的排序设置，使用【焊缝组排序】按钮进行选中工序内的焊缝组自动焊接顺序进行排序。



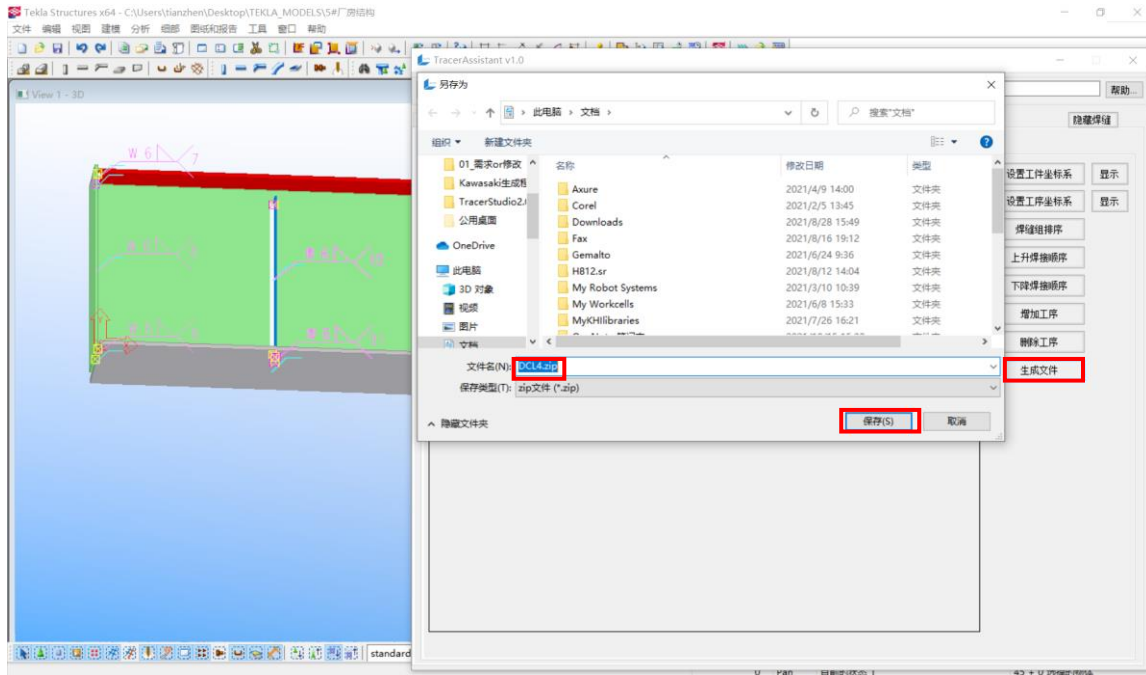
2.6.5 增加、删除、修改

根据实际加工的工序安排，对工序表进行增加、删除和修改。



2.6.6 生成文件

使用【生成文件】按钮生成选中构件的焊缝数据，进行导出。选择需要存储的路径及对应的名称。默认名称为构建编号。



生成的数据存储为 zip 压缩包形式，直接在 TracerStudio 上进行使用。

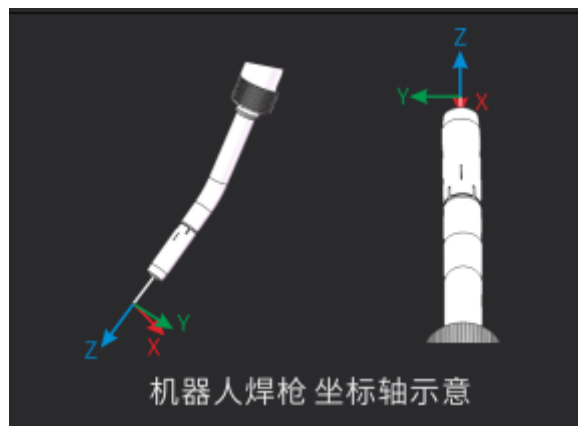
3. TracerStudio 程序编辑

3.1 准备工作

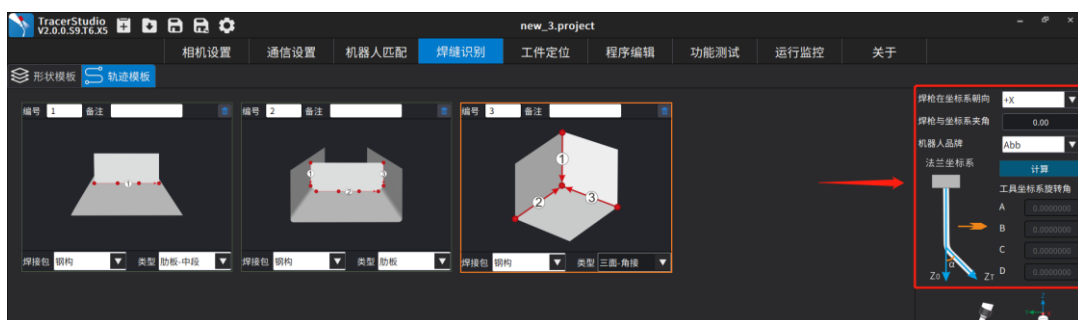
使用本功能前，请完成以下内容：

- 1) 完成机器人 TCP 标定；
- 2) 完成机器人焊枪工具坐标系旋转量设定；
- 3) 使用 TracerCalibration 完成相机内参标定；
- 4) 使用 TracerCalibration 完成手眼标定。

其中，要求工具坐标系的方向如图所示：



可使用 TracerStudio→【焊缝识别】→【轨迹模板】中的工具进行计算。



- 判断方向，测量角度：判断焊枪在法兰坐标系下的朝向（+X\ -X\ +Y\ -Y）四种，测量焊枪弯管与法兰 Z 轴的夹角。

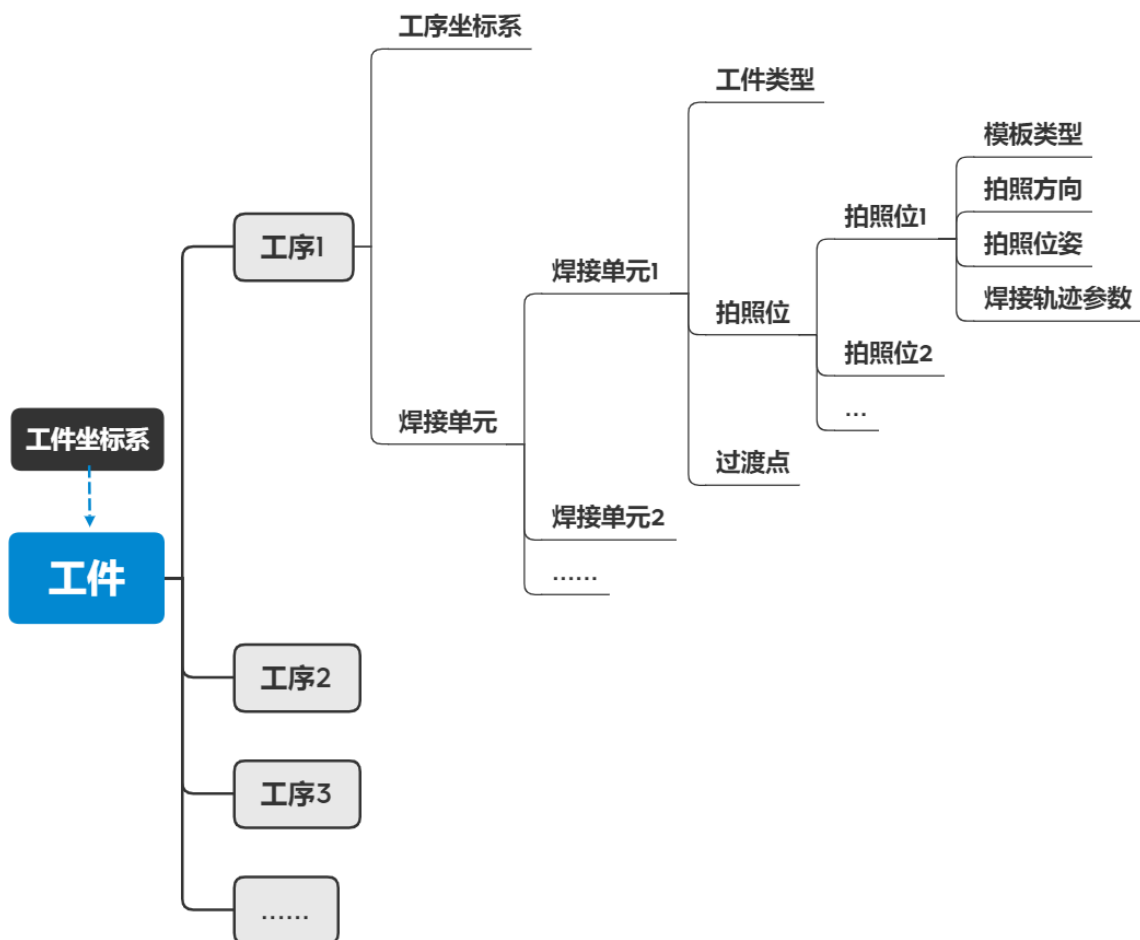
- 计算结果：在软件中输入参数后，点击计算，将结果写入机器人工具设定参数。
 - 结果判断：将坐标系切换至工具坐标系，分别移动 X\Y\Z 轴，确认方向。
- ⚠ 注意：工具坐标系旋转量若修改，必须重新进行手眼标定。

3.2 功能介绍

【程序编辑】功能是通过导入焊缝信息，自动计算生成机器人程序的，结合焊缝识别功能，同时实现了机器人自动编程以及结合视觉引导的自动焊接。

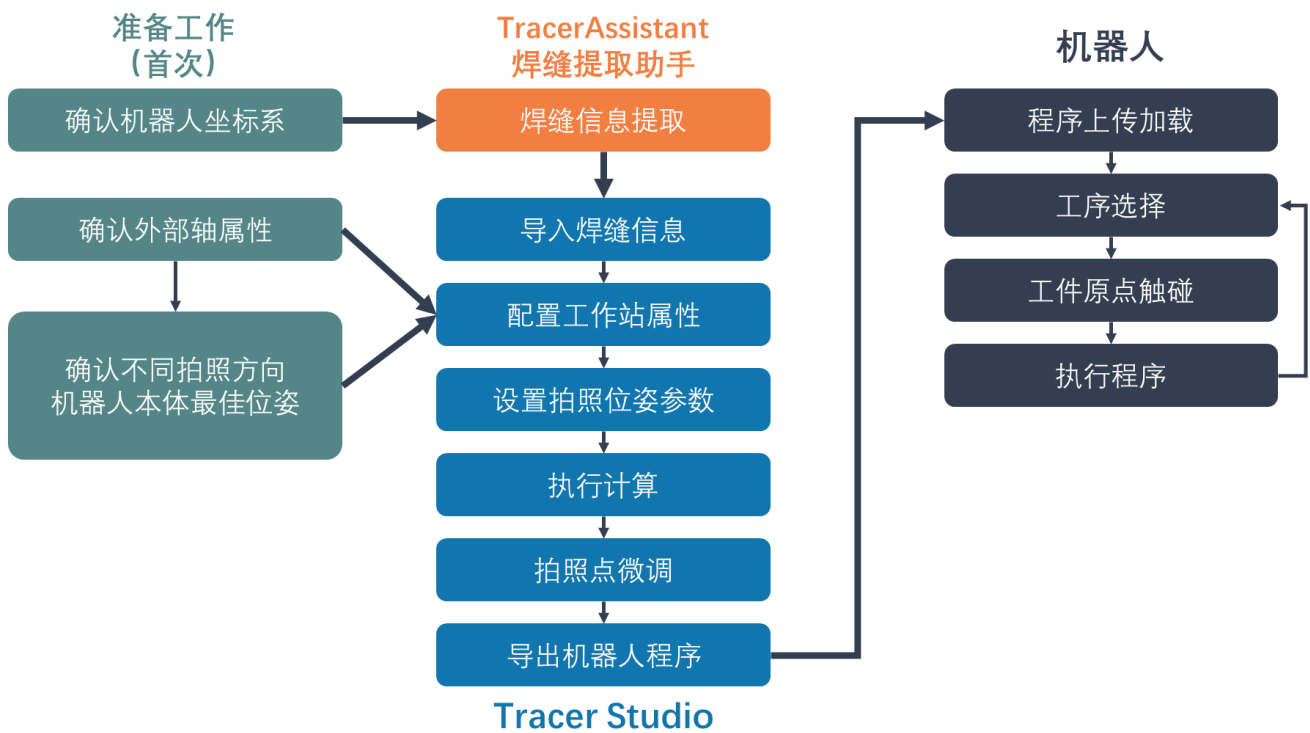
【程序编辑】以工件为单元，按照加工时工件装夹或放置方式分为不同的工序，每个工序分为若干个焊接单元。每个焊接单元所实现的过程为焊接轨迹计算。

3.2.1 数据结构

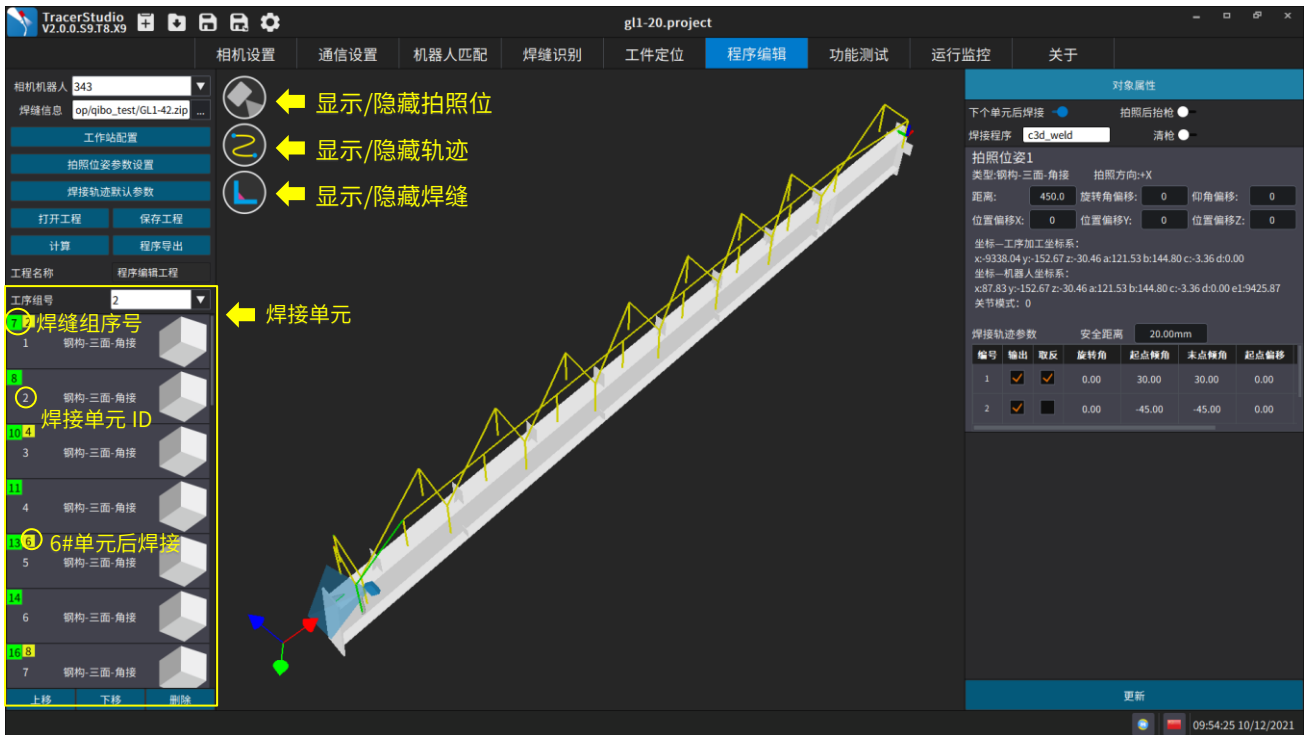


3.2.2 执行流程

整体流程如下图所示，需要先确认工作站的基本硬件构成，以及拍照时的最优位姿，然后 TracerStudio 进行操作。其中工作站配置、拍照位姿参数对于同样的工作站只需要配置一次即可。



3.3 参数配置



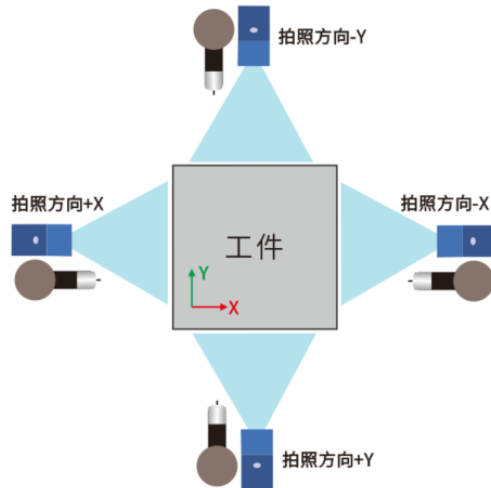
3.3.1 配置工作站属性



需要配置的信息如图所示。包括机器人品牌、安装方式、外部轴数量、是否联动以及安全高度。此外还需要配置外部轴属性、不同拍照方向的关节模式、机器人本体坐标参考值。

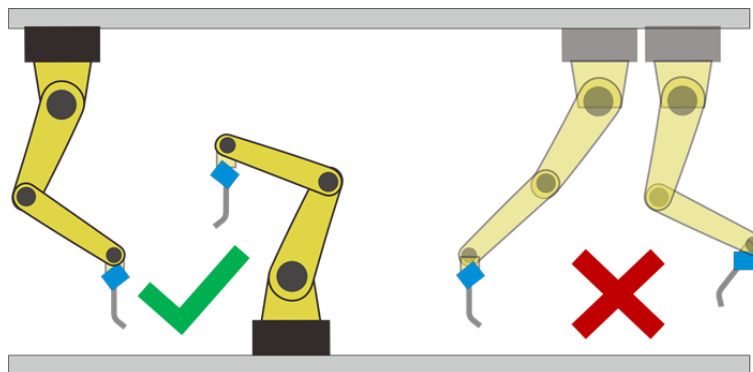
参数名称	属性	说明
机器人品牌	/	选择对应的机器人品牌，则会生成对应品牌的机器人程序
安装方式	/	正装 / 倒装
安全高度	/	基于焊线最高点的抬高距离
清枪程序	/	清枪子程序的名称，程序需要用户编写
分组阈值	/	拍照位距离小于阈值的会分为一组，同一组会先集中拍照后集中焊接
Tekla 修正功能	/	开启/关闭 Tekla 修正功能
外部轴数量	/	外部轴数量
是否联动	/	位置坐标基于本体为不联动 位置坐标基于包含外部轴的世界坐标系为联动
输出基础库程序	/	通信程序、焊接程序等 C3D 机器人基础库是否输出
拍照方向 — 六轴机器人关节模式	拍照方向	六轴机器人关节模式
	+X	拍照方向为+X 时，机器人的关节模式
	-X	拍照方向为-X 时，机器人的关节模式
	+Y	拍照方向为+Y 时，机器人的关节模式
	-Y	拍照方向为-Y 时，机器人的关节模式
外部轴属性	属性	直线外部轴的属性选项为 +X / -X / +Y / -Y / +Z / -Z
	聚类阈值	拍照位的外部轴差异小于阈值时，使其一致 减少外部轴的运动
拍照方向 — 机器人本体坐标参考值	拍照方向	机器人本体坐标参考值
	+X	拍照方向为+X 时，机器人本体对应轴的坐标
	-X	拍照方向为-X 时，机器人本体对应轴的坐标
	+Y	拍照方向为+Y 时，机器人本体对应轴的坐标
	-Y	拍照方向为-Y 时，机器人本体对应轴的坐标

● 拍照方向 — 六轴机器人关节模式



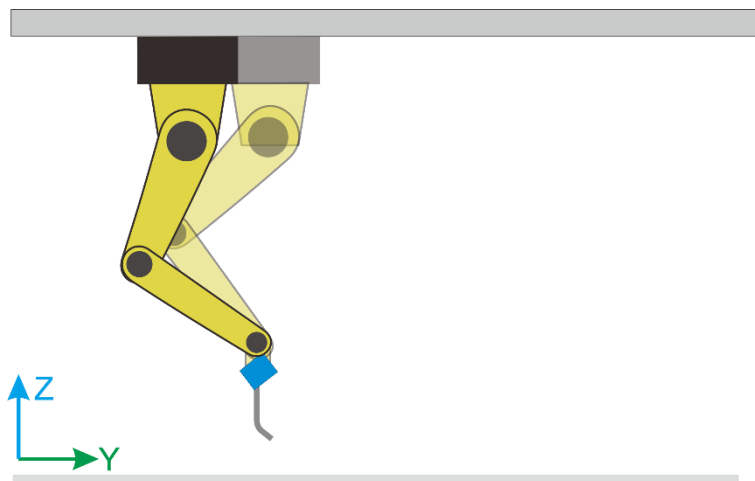
机器人的点位不仅需要定义坐标值和姿态，还需要对关节模式进行定义，不然会存在多个解。每个品牌的关节模式参数含义及数量均不同，需要自行查阅。决定关节模式的是拍照方向，因此需要设置不同拍照方向下的关节模式。

关节模式需要通过示教来得到，示教时使机器人处于比较合适的姿态。



- 拍照方向 — 机器人本体坐标参考值

对于有外部轴的情况，一个坐标对应无数个解，需要定义对应轴方向的机器人本体值才能确定点位的外部轴坐标，而机器人的本体值和拍照方向相关，因此需要配置不同拍照方向下，对应轴方向机器人本体坐标的参考值。



示例：

拍照方向 +X
CONFIG : R U N
X=156.57mm, Y=32.65mm, Z=-576.32mm

拍照方向 -X
CONFIG : R U F
X=356.72, Y=-17.56mm, Z =-578.78mm

拍照方向 +Y
CONFIG : R D N
X=16.75mm, Y=-352.75mm, Z=-586.54mm

拍照方向 -Y
CONFIG : L D F
X=15.78mm, Y=268.38mm, Z=-582.38mm

工作站配置

机器人品牌: Fanuc 安装方式: 倒装
外部轴: 2 是否联动: 是
安全高度: 100.00 mm 输出基础库程序: 是

拍照方向--六轴机器人关节模式				
编号	+X	-X	+Y	-Y
1	R	R	R	L
2	U	U	D	D
3	N	F	N	F

外部轴属性				拍照方向--机器人本体坐标参考值			
外部轴	属性	最大行程	聚类阈值	+X	-X	+Y	-Y
1	+X	5000.00	200.00	156.57	356.72	16.75	15.78
2	+Y	5000.00	200.00	32.65	-17.56	-352.75	268.38

应用 确定

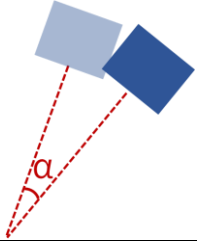
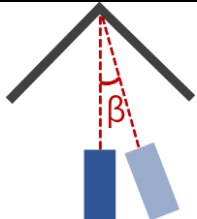
注意：数据及配置仅为示例，不可参考
示例机器人为FANUC，其他机器人参数不同

3.3.2 设置拍照位参数

相机拍照位参数设置

最佳拍照距离: 400.00 mm 最远拍照距离: 600.00 mm
有效视场系数: 1.00 位置偏移X: 0 mm
仰角偏移: 0 mm 位置偏移Y: 0 mm
旋转角偏移: 0 mm 位置偏移Z: 0 mm

应用 确定

参数名称	说明	
最佳拍照距离	相机相对于工件的拍照距离的最佳值	
最远拍照距离	相机相对于工件的拍照距离的最远值	
有效视场系数	视场范围*有效视场系数为有效区域	
仰角偏移	相机光轴与水平面的夹角为仰角，夹角由计算得到，偏移量为叠加量	
旋转角偏移	相机光轴投影与构件投影的角平分线夹角为旋转角，旋转角由计算得到，偏移量为叠加量	

3.3.3 修改焊接轨迹默认参数

焊接包 钢构

焊接类型 肋板

拍照后抬焊 ☐

焊接程序 c3d_weld

安全距离 20.00mm

编号	输出	取反	姿态旋转角	起点倾角	终点倾角	起点偏移	末点偏移
1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0.00	30.00	30.00	0.00	0.00
2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0.00	45.00	-45.00	0.00	0.00
3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0.00	30.00	30.00	0.00	0.00

更新

全部更新

应用

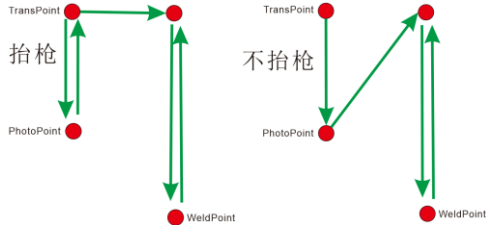
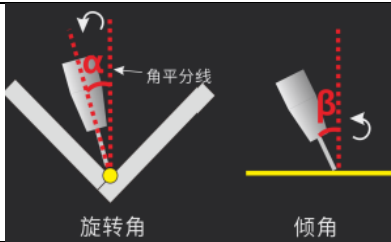

全部应用

选择【焊接包】与【焊接类型】，可配置当前焊接类型的参数。操作说明如下：

操作	说明
更新	将当前类型的当前选中参数更新至工程的对象属性中
全部更新	将所有类型的当前选中参数更新至工程的对象属性中
应用	将当前类型的当前选中参数应用并保存，对象属性不更新

全部应用	将所有类型的当前选中参数应用并保存，对象属性不更新
------	---------------------------

可配置的参数如下：

参数	说明	备注
拍照后抬枪	拍照后是否移动至安全高度	
焊接程序	焊接程序的名称	Todo
轨迹参数	安全距离	焊接轨迹的安全距离
	姿态旋转角 起点倾角 末点倾角	
	起点偏移 末点偏移	

3.4 操作流程

3.4.1 焊缝信息导入

焊缝信息是通过 Tekla 软件中的 TracerAssistant 焊缝提取助手插件导出。

3.4.2 执行计算

点击 **计算** 按钮，软件会生成工序以及所有焊接单元。

3.4.3 焊接单元编辑

● 顺序调整

点击 **上移**、**下移** 可以调整单元顺序。

下个单元后焊接 ☒
 拍照后抬枪 ☐

焊接程序
 清枪 ☐

拍照位姿1
 类型:钢构-三面-角接 拍照方向:+X
 距离: 旋转角偏移: 仰角偏移:
 位置偏移X: 位置偏移Y: 位置偏移Z:
 坐标—工序加工坐标系:
 x:-9338.04 y:-152.67 z:-30.46 a:121.53 b:144.80 c:-3.36 d:0.00
 坐标—机器人坐标系:
 x:87.83 y:-152.67 z:-30.46 a:121.53 b:144.80 c:-3.36 d:0.00 e:9425.87
 关节模式: 0

焊接轨迹参数 安全距离

编号	输出	取反	旋转角	起点倾角	末点倾角	起点偏移
1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0.00	30.00	30.00	0.00
2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0.00	-45.00	-45.00	0.00

● 程序调整

可单独修改选中单元的【拍照后抬枪】、【焊接程序】、【清枪】。

● 分组调整

设置【下个单元后焊接】，若开启，则表示和下个单元分为了一组。

● 拍照位姿调整

可单独调整拍照位置的计算参数。

⚠ 注意：上述参数影响生成的机器人程序，若修改，需要重新生成程序。

● 焊接轨迹参数调整

可单独调整焊接轨迹参数的参数。 **⚠ 注意：**本参数更新后无需重新生成机器人程序。

3.4.4 程序导出

点击 程序导出，导出机器人程序。

3.4.5 工程保存

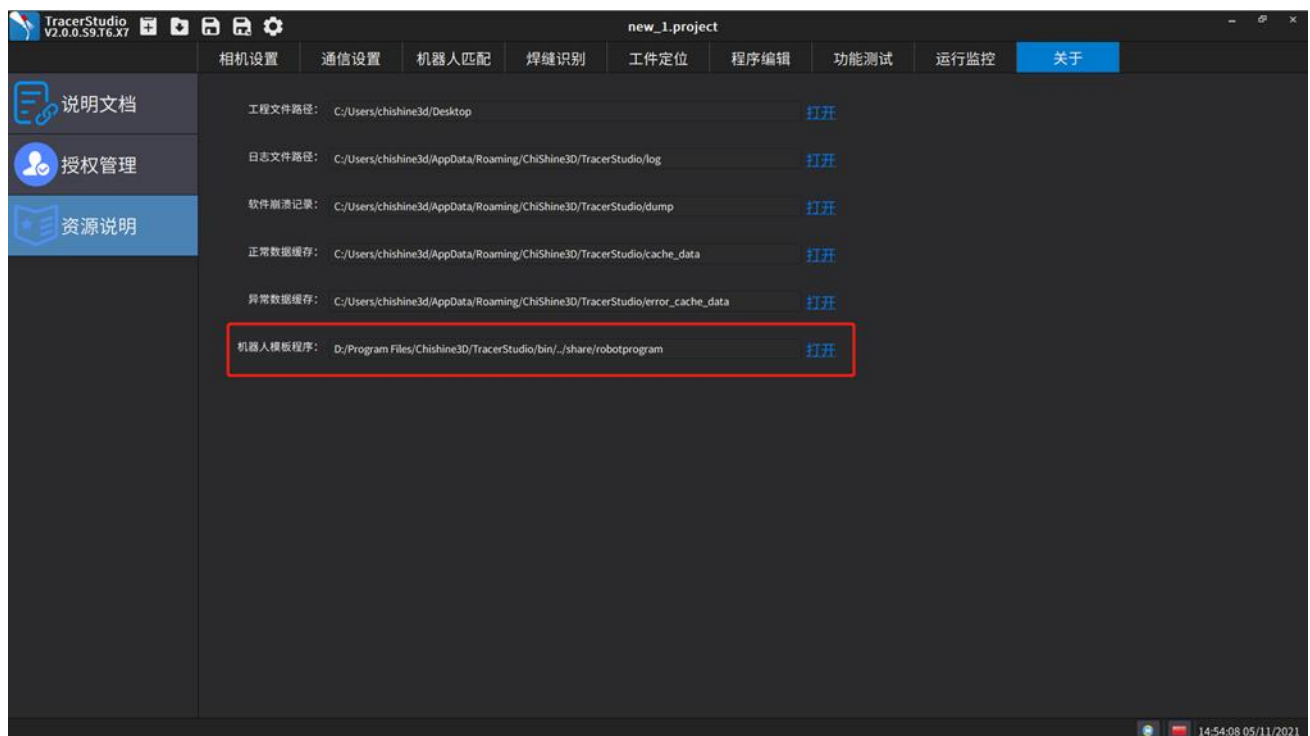
点击 **保存工程**，保存本工程。当需要修改的时候，可以导入工程。

3.5 程序模板修改

生成的机器人程序中有固定的代码，例如通信代码、工具名称、焊接程序等，固定代码可以通过修改文件内容实现永久修改，而不用每次生成后重复修改。

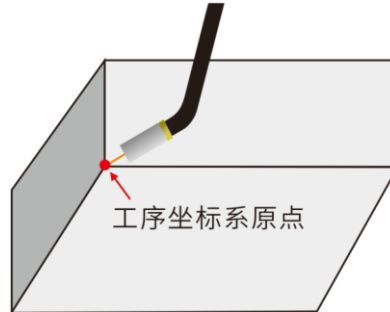
若代码为独立的文件，则修改对应的文件；若代码为部分代码段，修改对应的 json 文件。

文件路径：



4. 机器人控制

1) 机器人原点示教



TracerStudio 生成出的程序不可以直接运行，第一步首先是进行坐标系对齐，控制机器人将 TCP（焊枪端点）移动至工序坐标系原点，并示教记录。

2) 修改程序

根据需求，可以示教整个工序的起始位置、结束位置、清枪点等。

3) 试运行

不开启焊接的情况下，手动运行程序，对于拍照点位不合适、点位不可达等问题进行预先判断，并采取相应的处理措施。

4) 运行程序

打开焊接指令，自动运行程序。

附录 A 机器人程序说明

A.1 FANUC

● Process 主程序

```
1: UFRAME_NUM=1 ;
2: UTOOL_NUM=1 ;
3://L P[99] 100mm/sec FINE ;
4: PR[99]=P[99] ;
5: R[50]=0 ;
6: CALL C3D_REFOFFS ;
7: OFFSET CONDITION PR[100] ;
8: CALL C3D('C1', 'CONNECT') ; //建立连接
9: CALL C3D('C1', 'REQUEST', 0, 1) ; //启动视觉服务
10: CALL C3D('C1', 'REQUEST', 20, 1, 2) ; //选择工序
11: --eg:UNIT 1 ; //焊接单元1
12:J P[1] 30% FINE Offset ; //焊接单元1_过渡点
13:L P[2] 300mm/sec FINE Offset ; //焊接单元1_拍照点
14: CALL C3D('C1', 'REQUEST', 21, 1, 1) ; //请求数据
15://L P[3] 300mm/sec FINE Offset ; //焊接单元1_拍照点, 如果需要抬枪, 取消注释
16: CALL C3D_WELD ;
17: --eg:UNIT 2 ;
18:J P[4] 30% FINE Offset ;
19:L P[5] 300mm/sec FINE Offset ;
20: CALL C3D('C1', 'REQUEST', 21, 2, 1) ;
21://L P[6] 300mm/sec FINE Offset ;
22: CALL C3D_WELD ;
23: --eg:UNIT 3 ;
24:J P[7] 30% FINE Offset ;
25:L P[8] 300mm/sec FINE Offset ;
26: CALL C3D('C1', 'REQUEST', 21, 3, 1) ;
27://L P[9] 300mm/sec FINE Offset ;
28: CALL C3D_WELD ;
22: CALL C3D('C1', 'REQUEST', 20, 0, 1) ; //关闭工序
23: CALL C3D('C1', 'CLOSE') ; //断开连接
```

● C3D_WELD 子程序 (可在程序模板中修改)

```
1: IF (R[11]<>0) THEN ;
2: PAUSE ;
3: ENDIF ;
```

```
4: R[49]=1 ;
5: PR[50]=PR[R[49]] ;
6: PR[50,3]=R[50] ;
7:L PR[50] 600mm/sec FINE ;
8: FOR R[41]=1 TO R[13] ; //焊缝逐条焊接
9: R[42]=R[41]+13 ;
10: R[43]=R[41]+20 ;
11: R[44]=R[41]+30 ;
12:L PR[R[49]] 600mm/sec FINE ;
13: R[49]=R[49]+1 ;
14: IF (R[R[43]]>5) THEN ;
15: R[49]=R[49]+R[R[42]] ;
16: JMP LBL[1] ;
17: ENDIF ;
18:L PR[R[49]] 600mm/sec FINE ;
19: R[49]=R[49]+1 ;
20: --eg: Flat weld(Ping) ;
21: IF (R[R[44]]=0) THEN ;
22: Weld Start[1,1] ; //平焊缝的工艺
23: R[46]=20 ;
24: ENDIF ;
25: --eg: Stand weld(Li) ;
26: IF (R[R[44]]=1) THEN ;
27: Weld Start[1,2] ; //立焊缝的工艺
28: R[46]=10 ;
29: ENDIF ;
30: IF (R[R[43]]<2) THEN ;
31: Weave Sine[1] ;
32: ELSE ;
33: IF (R[R[43]]<5) THEN ;
34: Weave Sine[2] ;
35: ENDIF ;
36: FOR R[45]=2 TO R[R[42]] ;
37:L PR[R[49]] R[46]mm/sec FINE ;
38: R[49]=R[49]+1 ;
39: ENDFOR ;
40: Weave End[1] ;
41: Weld End[1,1] ;
42: LBL[1] ;
43:L PR[R[49]] 600mm/sec FINE ;
44: WAIT 100.00(sec) ;
45: R[49]=R[49]+1 ;
```

```
46: ENDFOR ;  
47: PR[50]=PR[R[49]] ;  
48: PR[50,3]=R[50] ;  
49:L PR[50] 600mm/sec FINE ;
```

A.2 Kawasaki

● Process 主程序

```
. PROGRAM process_3()  
BASE world  
TOOL tool_c3d  
LMOVE p_home //机器人原点  
JMOVE p0_3 //工序加工坐标系原点，必须进行示教  
DELAY 0.1  
DELAY 0.1  
BREAK  
safe_z = 0  
CALL c3d_refoffs_3 //执行数据校准  
CALL c3d_connect //请求连接  
CALL c3d_request(0, 1, 1) //打开运行监控  
DELAY 0.1  
DELAY 0.1  
BREAK  
CALL c3d_request(20, 1, 3) //选择工序  
JMOVE start_3[1] //焊接单元1_起始过渡点  
LMOVE photo_3[1, 1] //焊接单元1_拍照点1  
DELAY 0.1  
DELAY 0.1  
BREAK  
CALL c3d_request(21, 1, 1) //请求数据  
; LMOVE end_3[1]  
CALL c3d_weld //执行焊接  
JMOVE start_3[2]  
LMOVE photo_3[2, 1]  
DELAY 0.1  
DELAY 0.1  
BREAK  
CALL c3d_request(21, 2, 1)  
LMOVE photo_3[2, 2]  
DELAY 0.1  
DELAY 0.1
```

```
BREAK
CALL c3d_request(21, 2, 2)
; LMOVE end_3[2]
CALL c3d_weld
LMOVE end_3[2]
BREAK
CALL c3d_request(20, 0, 3) //关闭工序
.END
```

● C3D_WELD 子程序 (可在程序模板中修改)

```
.PROGRAM c3d_weld()
POINT .safe_point = start_weld[1]
POINT safez = TRANS(0, 0, safe_z, 0, 0, 0)
POINT/Z safe_point = safez
LMOVE .safe_point
FOR .i = 1 TO path_num
    LMOVE start_weld[.i]
    IF weld_type[.i] == 0 THEN
        IF weld_gap > 5 THEN
            W1SET 20 = 17, 150, 17, 6.5, 0.8, 0
        END
        IF 2 < weld_gap < 5 THEN
            W1SET 20 = 17, 150, 17, 6.5, 0.8, 0
        END
        IF weld_gap < 2 THEN
            W1SET 20 = 17, 150, 17, 6.5, 0.8, 0
        END
    END
    IF weld_type[.i] == 1 THEN
        IF weld_gap > 5 THEN
            W1SET 20 = 17, 150, 17, 6.5, 0.8, 0
        END
        IF 2 < weld_gap < 5 THEN
            W1SET 20 = 17, 150, 17, 6.5, 0.8, 0
        END
        IF weld_gap < 2 THEN
            W1SET 20 = 17, 150, 17, 6.5, 0.8, 0
        END
    END
    LWS weld_points[.i, 1]
    FOR .j = 2 TO weld_point_num[.i] - 1
```

```
LWC weld_points[. i, . j] 20
END
.m = weld_point_num[. i]
LWE weld_points[. i, . m], 20, 1
LMOVE end_weld[. i]
END
POINT .safe_point = end_weld[path_num]
POINT safez = TRANS(0, 0, safe_z, 0, 0, 0)
POINT/Z safe_point = safez
LMOVE .safe_point
.END
```

A.3 STEP

● Process 主程序

```
Tool(tool_C3D);
PTP(Phome, NULL, NULL, tool_C3D, WORLD);
//建立连接并开启运行监控
ProgCall(C3D_Connect);
safe_z := 0;
CARPOSEXTSet(0, P0_1); //工序加工坐标系原点，必须进行示教
SaveData(P0_1);
P0 := P0_1;
ProgCall(C3D_RefOffs); //执行数据校准
messageNum := 20; //选择工序
commandNum := 1;
processNum := 1;
ProgCall(C3D_Request);
PTP(startTransPoint_1_1, NULL, NULL, tool_C3D, WORLD); //焊接单元1_起始过渡点
Lin(photoPoint_1_1_1, NULL, NULL, NULL, tool_C3D, WORLD); //焊接单元1_拍照点1
messageNum := 21;
weldUnitNum := 1;
serialNum := 1;
ProgCall(C3D_Request); //请求数据
// Lin(endTransPoint_1_1, NULL, NULL, NULL, tool_C3D, WORLD); //拍照结束抬枪
ProgCall(C3D_Weld); //执行焊接
PTP(startTransPoint_1_2, NULL, NULL, tool_C3D, WORLD);
Lin(photoPoint_1_2_1, NULL, NULL, NULL, tool_C3D, WORLD); //焊接单元2_拍照点1
messageNum := 22;
weldUnitNum := 2;
serialNum := 1;
```

```
ProgCall (C3D_Request);  
Lin(photoPoint_1_2_2, NULL, NULL, NULL, tool_C3D, WORLD); //焊接单元2_拍照点2  
messageNum := 22;  
weldUnitNum := 2;  
serialNum := 2;  
ProgCall (C3D_Request);  
// Lin(endTransPoint_1_2, NULL, NULL, NULL, tool_C3D, WORLD); //拍照结束抬枪  
ProgCall (C3D_Weld);  
messageNum := 20;  
commandNum := 0;  
ProgCall (C3D_Request);
```

- C3D_WELD 子程序 (可在程序模板中修改)

```
Tool (tool_C3D);  
DOSet (D09, FALSE); //关闭相机防护装置, 信号需要现场定制  
i:=1;  
WHILE (i<=pathNum) DO //第几条焊线判断, 循环处理每条焊线  
pathSerial := i;  
WaitIsFinished();  
messageNum := 10000;  
WaitIsFinished();  
ProgCall (C3D_Request);  
IF (i==1) THEN  
safe_point:=CARTPOSEXTRead(1);  
safe_point.Z:=safe_z;  
PTP(safe_point, NULL, NULL, NULL, tool_C3D, WORLD);  
END_IF  
i:=i+1;  
IF (read_step==1080) THEN //获取焊缝信息完成  
Lin(transStart, NULL, NULL, NULL, tool_C3D, WORLD); //运动至焊缝起始过渡点  
IF (weldType==0) THEN //焊缝类型判断, 0平焊缝, 1立焊缝。并进行焊接参数定制  
IF (gap<6) THEN  
C3D_arcMode:=arcMode_1;  
C3D_arcon:=arcondata_1;  
C3D_arcdata:=arcdata_1;  
C3D_weave:=weavedata_1;  
C3D_track:=trackdata_1;  
C3D_arcoff:=arcoffdata_1;  
ELSIF (gap>=6) THEN  
C3D_arcMode:=arcMode_2;  
C3D_arcon:=arcondata_2;  
C3D_arcdata:=arcdata_2;
```

```
C3D_weave:=weavedata_2;
C3D_track:=trackdata_2;
C3D_arcoff:=arcoffdata_2;
END_IF
ELSIF (weldType==1) THEN
  IF (gap<6) THEN
    C3D_arcMode:=arcMode_3;
    C3D_arcon:=arcondata_3;
    C3D_arcdata:=arcdata_3;
    C3D_weave:=weavedata_3;
    C3D_track:=trackdata_3;
    C3D_arcoff:=arcoffdata_3;
  ELSEIF (gap>=6) THEN
    C3D_arcMode:=arcMode_4;
    C3D_arcon:=arcondata_4;
    C3D_arcdata:=arcdata_4;
    C3D_weave:=weavedata_4;
    C3D_track:=trackdata_4;
    C3D_arcoff:=arcoffdata_4;
  END_IF
END_IF
ARCMODE (C3D_arcMode);
Lin(weldPoint1, NULL, NULL, NULL, tool_C3D, WORLD); //焊缝起始点
ARCON (C3D_arcon, C3D_arcdata);
SWITCH (weldPointNum) //焊缝点数测试, 根据不同的点数进行焊缝分支选择
CASE 0:
  Message("wrong", eInfo);
CASE 2:
  WLin(weldPoint2, NULL, NULL, NULL, C3D_weave, C3D_track, tool_C3D, WORLD);
CASE 3:
  WLin(weldPoint2, NULL, Z0, NULL, C3D_weave, C3D_track, tool_C3D, WORLD);
  WLin(weldPoint3, NULL, NULL, NULL, C3D_weave, C3D_track, tool_C3D, WORLD);
CASE 4:
  WLin(weldPoint2, NULL, Z0, NULL, C3D_weave, C3D_track, tool_C3D, WORLD);
  WLin(weldPoint3, NULL, Z0, NULL, C3D_weave, C3D_track, tool_C3D, WORLD);
  WLin(weldPoint4, NULL, NULL, NULL, C3D_weave, C3D_track, tool_C3D, WORLD);
CASE 5:
  WLin(weldPoint2, NULL, Z0, NULL, C3D_weave, C3D_track, tool_C3D, WORLD);
  WLin(weldPoint3, NULL, Z0, NULL, C3D_weave, C3D_track, tool_C3D, WORLD);
  WLin(weldPoint4, NULL, Z0, NULL, C3D_weave, C3D_track, tool_C3D, WORLD);
  WLin(weldPoint5, NULL, NULL, NULL, C3D_weave, C3D_track, tool_C3D, WORLD);
CASE 6:
```

```
WLin(weldPoint2, NULL, Z0, NULL, C3D_weave, C3D_track, tool_C3D, WORLD);
WLin(weldPoint3, NULL, Z0, NULL, C3D_weave, C3D_track, tool_C3D, WORLD);
WLin(weldPoint4, NULL, Z0, NULL, C3D_weave, C3D_track, tool_C3D, WORLD);
WLin(weldPoint5, NULL, Z0, NULL, C3D_weave, C3D_track, tool_C3D, WORLD);
WLin(weldPoint6, NULL, NULL, NULL, C3D_weave, C3D_track, tool_C3D, WORLD);
CASE 7:
WLin(weldPoint2, NULL, Z0, NULL, C3D_weave, C3D_track, tool_C3D, WORLD);
WLin(weldPoint3, NULL, Z0, NULL, C3D_weave, C3D_track, tool_C3D, WORLD);
WLin(weldPoint4, NULL, Z0, NULL, C3D_weave, C3D_track, tool_C3D, WORLD);
WLin(weldPoint5, NULL, Z0, NULL, C3D_weave, C3D_track, tool_C3D, WORLD);
WLin(weldPoint6, NULL, Z0, NULL, C3D_weave, C3D_track, tool_C3D, WORLD);
WLin(weldPoint7, NULL, NULL, NULL, C3D_weave, C3D_track, tool_C3D, WORLD);
CASE 8:
WLin(weldPoint2, NULL, NULL, NULL, C3D_weave, C3D_track, tool_C3D, WORLD);
WLin(weldPoint3, NULL, NULL, NULL, C3D_weave, C3D_track, tool_C3D, WORLD);
WLin(weldPoint4, NULL, NULL, NULL, C3D_weave, C3D_track, tool_C3D, WORLD);
WLin(weldPoint5, NULL, NULL, NULL, C3D_weave, C3D_track, tool_C3D, WORLD);
WLin(weldPoint6, NULL, NULL, NULL, C3D_weave, C3D_track, tool_C3D, WORLD);
WLin(weldPoint7, NULL, NULL, NULL, C3D_weave, C3D_track, tool_C3D, WORLD);
WLin(weldPoint8, NULL, NULL, NULL, C3D_weave, C3D_track, tool_C3D, WORLD);
END_SWITCH
ARCOFF (C3D_arcoff);
Lin(transEnd, NULL, NULL, NULL, tool_C3D, WORLD); //运行至焊缝末点过渡点
END_IF
END_WHILE
safe_point:=CARTPOSEXTRead(10);
safe_point.Z:=safe_z;
Lin(safe_point, NULL, NULL, NULL, tool_C3D, WORLD);
DOSet(D09, TRUE); //打开防护装置, 需要现场自行定制
//Lin(PVission, NULL, NULL, NULL, tool_C3D, WORLD);
```