

10. 机器人力控

10.1. 获取力传感器配置

20-ATI 传感器, 21-中科米点, 22-伟航敏芯;device 设备号, 坤维 (0-KWR75B), 航天十一院 (0-MC

10.1.1. 代码示例

```
1  from fairino import Robot
2  import time
3  # 与机器人控制器建立连接, 连接成功返回一个机器人对象
4  robot = Robot.RPC('192.168.58.2')
5  company = 17      #传感器厂商, 17-坤维科技
6  device = 0        #传感器设备号
7  error = robot.FT_SetConfig(company, device) #配置力传感器
8  print("配置力传感器错误码",error)
9  config = robot.FT_GetConfig() #获取力传感器配置信息
10 print('获取力传感器配置信息',config)
11 time.sleep(1)
12 error = robot.FT_Activate(0) #传感器复位
13 print("传感器复位错误码",error)
14 time.sleep(1)
15 error = robot.FT_Activate(1) #传感器激活
16 print("传感器激活错误码",error)
17 time.sleep(1)
18 error = robot.SetLoadWeight(0.0) #末端负载设置为零
19 print("末端负载设置为零错误码",error)
20 time.sleep(1)
21 error = robot.SetLoadCoord(0.0,0.0,0.0) #末端负载质心设置为零
22 print("末端质心设置为零错误码",error)
23 time.sleep(1)
24 error = robot.FT_SetZero(0) #传感器去除零点
25 print("传感器去除零点错误码",error)
26 time.sleep(1)
27 error = robot.FT_GetForceTorqueOrigin() #查询传感器原始数据
28 print("查询传感器原始数据",error)
29 error = robot.FT_SetZero(1) #传感器零点矫正, 注意此时末端不能安装工具, 只有力传感器
30 print("传感器零点矫正",error)
31 time.sleep(1)
32 error = robot.FT_GetForceTorqueRCS() #查询传感器坐标系下数据
33 print("查询传感器坐标系下数据",error)
```

10.2. 力传感器配置

在 Python 版本发生变更: SDK-v2.0.8-3.7.8

ATI传感器, 21-中科米点, 22-伟航敏芯, 23-NBIT, 24-鑫精诚(XJC), 26-NSR;
A-200-4), ATI(0-AXIA80-M8), 中科米点(0-MST2010), 伟航敏芯(0-WHC6L-YB-10A), NBIT(0-

10.3. 力传感器激活

原型	<code>FT_Activate(state)</code>
描述	力传感器激活
必选参数	<ul style="list-style-type: none"><code>state</code>: 0-复位, 1-激活
默认参数	无
返回值	错误码 成功-0 失败- errcode

10.4. 力传感器校零

原型	<code>FT_SetZero(state)</code>
描述	力传感器校零
必选参数	<ul style="list-style-type: none"><code>state</code>: 0-去除零点, 1-零点矫正
默认参数	无
返回值	错误码 成功-0 失败- errcode

10.5. 设置力传感器参考坐标系

在 python 版本发生变更: SDK-v2.0.5

原型	<code>FT_SetRCS(ref, coord=[0,0,0,0,0,0])</code>
描述	设置力传感器参考坐标系

必选参数	<ul style="list-style-type: none"> <code>ref</code>: 0-工具坐标系, 1-基坐标系
默认参数	<ul style="list-style-type: none"> <code>coord</code>: [x,y,z,rx,ry,rz]自定义坐标系值,默认[0,0,0,0,0,0]
返回值	错误码 成功-0 失败- errcode

10.5.1. 代码示例

```

1  from fairino import Robot
2  import time
3  # 与机器人控制器建立连接, 连接成功返回一个机器人对象
4  robot = Robot.RPC('192.168.58.2')
5  #负载辨识, 此时末端安装要辨识的工具, 工具安装在力传感器下方, 末端竖直向下
6  error = robot.FT_SetRCS(0)    #设置参考坐标系为工具坐标系, 0-工具坐标系, 1-基坐标系
7  print('设置参考坐标系错误码',error)
8  time.sleep(1)
9  tool_id = 10    #传感器坐标系编号
10 tool_coord = [0.0,0.0,35.0,0.0,0.0,0.0]    # 传感器相对末端法兰位姿
11 tool_type = 1    # 0-工具, 1-传感器
12 tool_install = 0 # 0-安装末端, 1-机器人外部
13 error = robot.SetToolCoord(tool_id,tool_coord,tool_type,tool_install)    #设置传
    感器坐标系, 传感器相对末端法兰位姿
14 print('设置传感器坐标系错误码',error)
15 time.sleep(1)
16 error = robot.FT_PdIdenRecord(tool_id)    #记录辨识数据
17 print('记录负载重量错误码',error)
18 time.sleep(1)
19 error = robot.FT_PdIdenRecord()    #计算负载重量, 单位kg
20 print('计算负载重量错误码',error)
21 #负载质心辨识, 机器人需要示教三个不同的姿态, 然后记录辨识数据, 最后计算负载质心
22 robot.Mode(1)
23 ret = robot.DragTeachSwitch(1)    #机器人切入拖动示教模式, 必须在手动模式下才能切入拖动示教
    模式
24 time.sleep(5)
25 ret = robot.DragTeachSwitch(0)
26 time.sleep(1)
27 error = robot.FT_PdCogIdenRecord(tool_id,1)
28 print('负载质心1错误码',error)#记录辨识数据
29 ret = robot.DragTeachSwitch(1)    #机器人切入拖动示教模式, 必须在手动模式下才能切入拖动示教
    模式
30 time.sleep(5)
31 ret = robot.DragTeachSwitch(0)
32 time.sleep(1)
33 error = robot.FT_PdCogIdenRecord(tool_id,2)
34 print('负载质心2错误码',error)
35 ret = robot.DragTeachSwitch(1)    #机器人切入拖动示教模式, 必须在手动模式下才能切入拖动示教
    模式
36 time.sleep(5)
37 ret = robot.DragTeachSwitch(0)
38 time.sleep(1)
39 error = robot.FT_PdCogIdenRecord(tool_id,3)
40 print('负载质心3错误码',error)
41 time.sleep(1)
42 error = robot.FT_PdCogIdenCompute()
43 print('负载质心计算错误码',error)

```

10.6. 负载重量辨识计算

原型	<code>FT_PdIdenCompute()</code>
描述	负载重量辨识计算
必选参数	无
默认参数	无
返回值	<ul style="list-style-type: none">错误码 成功-0 失败- errcode<code>weight</code>：负载重量，单位 kg

10.7. 负载重量辨识记录

原型	<code>FT_PdIdenRecord(tool_id)</code>
描述	负载重量辨识记录
必选参数	<ul style="list-style-type: none"><code>tool_id</code>：传感器坐标系编号，范围[0~14]
默认参数	无
返回值	错误码 成功-0 失败- errcode

10.8. 负载质心辨识计算

原型	<code>FT_PdCogIdenCompute()</code>
描述	负载质心辨识计算
必选参数	无
默认参数	无
返回值	<ul style="list-style-type: none">错误码 成功-0 失败- errcode<code>cog=[cogx, cogy, cogz]</code>：负载质心，单位 mm

10.9. 负载质心辨识记录

原型	<code>FT_PdCogIdenRecord(tool_id, index)</code>
描述	负载质心辨识记录
必选参数	<ul style="list-style-type: none"><code>tool_id</code>：传感器坐标系编号，范围[0~14];<code>index</code>：点编号[1~3]
默认参数	无
返回值	错误码 成功-0 失败- errcode

10.10. 获取参考坐标系下力/扭矩数据

原型	<code>FT_GetForceTorqueRCS()</code>
描述	获取参考坐标系下力/扭矩数据
必选参数	无
默认参数	无
返回值	<ul style="list-style-type: none">错误码 成功-0 失败- errcode<code>data=</code> <code>[fx,fy,fz,tx,ty,tz]</code> : 参考坐标系下力/扭矩数据

10.10.1. 代码示例

```
1 import frrpc
2 # 与机器人控制器建立连接, 连接成功返回一个机器人对象
3 robot = frrpc.RPC('192.168.58.2')
4 rcs = robot.FT_GetForceTorqueRCS() #查询传感器坐标系下数据
5 print(rcs)
```

10.11. 获取力传感器原始力/扭矩数据

原型	<code>FT_GetForceTorqueOrigin()</code>
描述	获取力传感器原始力/扭矩数据
必选参数	无
默认参数	无
返回值	<ul style="list-style-type: none">错误码 成功-0 失败- errcode<code>data=</code> <code>[fx,fy,fz,tx,ty,tz]</code> : 力传感器原始力/扭矩数据

10.11.1. 代码示例

```
1 import frrpc
2 # 与机器人控制器建立连接, 连接成功返回一个机器人对象
3 robot = frrpc.RPC('192.168.58.2')
4 origin = robot.FT_GetForceTorqueOrigin() #查询传感器原始数据
5 print(origin)
```



latest



10.12. 碰撞守护

原型	<code>FT_Guard(flag,sensor_num,select,force_torque,max_threshold,min_threshold)</code>
描述	碰撞守护
必选参数	<ul style="list-style-type: none"><code>flag</code> : 0-关闭碰撞守护, 1-开启碰撞守护;<code>sensor_num</code> : 力传感器编号;<code>select</code> : 六个自由度是否检测碰撞[fx,fy,fz,mx,my,mz], 0-不生效, 1-生效;<code>force_torque</code> : 碰撞检测力/力矩, 单位N或Nm;<code>max_threshold</code> : 最大阈值;<code>min_threshold</code> : 最小阈值;力/力矩检测范围:(force_torque-min_threshold,force_torque+max_threshold)
默认参数	无
返回值	错误码 成功-0 失败- errcode

10.12.1. 代码示例

```
1  from fairino import Robot
2  # 与机器人控制器建立连接, 连接成功返回一个机器人对象
3  robot = Robot.RPC('192.168.58.2')
4  #碰撞守护
5  actFlag = 1    #开启标志, 0-关闭碰撞守护, 1-开启碰撞守护
6  sensor_num = 1  #力传感器编号
7  is_select = [1,1,1,1,1,1]  #六个自由度选择[fx,fy,fz,mx,my,mz], 0-不生效, 1-生效
8  force_torque = [0.0,0.0,0.0,0.0,0.0,0.0]  #碰撞检测力和力矩, 检测范围 (force_torque-
min_threshold,force_torque+max_threshold)
9  max_threshold = [10.0,10.0,10.0,10.0,10.0,10.0]  #最大阈值
10 min_threshold = [5.0,5.0,5.0,5.0,5.0,5.0]  #最小阈值
11 P1=[-160.619,-586.138,384.988,-170.166,-44.782,169.295]
12 P2=[-87.615,-606.209,556.119,-102.495,10.118,178.985]
13 P3=[41.479,-557.243,484.407,-125.174,46.995,-132.165]
14 error = robot.FT_Guard(actFlag, sensor_num, is_select, force_torque,
max_threshold, min_threshold)  #开启碰撞守护
15 print("开启碰撞守护错误码",error)
16 error = robot.MoveL(P1,1,0)  #笛卡尔空间直线运动
17 print("笛卡尔空间直线运动错误码",error)
18 error = robot.MoveL(P2,1,0)
19 print("笛卡尔空间直线运动错误码",error)
20 error = robot.MoveL(P3,1,0)
21 print("笛卡尔空间直线运动错误码",error)
22 actFlag = 0
23 error = robot.FT_Guard(actFlag, sensor_num, is_select, force_torque,
max_threshold, min_threshold)  #关闭碰撞守护
24 print("关闭碰撞守护错误码",error)
```

10.13. 恒力控制

原型	<code>FT_Control(flag,sensor_num,select,force_torque,gain,adj_sign,ILC_sign,max_dis,max_ang)</code>
----	---

描述	恒力控制
必选参数	<ul style="list-style-type: none"><code>flag</code>：恒力控制开启标志，0-关，1-开；<code>sensor_num</code>：力传感器编号；<code>select</code>：六个自由度是否检测 [fx,fy,fz,mx,my,mz]，0-不生效，1-生效；<code>force_torque</code>：检测力/力矩，单位N或Nm；<code>gain</code>：[f_p,f_i,f_d,m_p,m_i,m_d],力PID参数，力矩PID参数；<code>adj_sign</code>：自适应启停状态，0-关闭，1-开启；<code>ILC_sign</code>：ILC控制启停状态，0-停止，1-训练，2-实操；<code>max_dis</code>：最大调整距离；<code>max_ang</code>：最大调整角度；
默认参数	无
返回值	错误码 成功-0 失败- errcode

10.13.1. 代码示例

```
1  from fairino import Robot
2  # 与机器人控制器建立连接，连接成功返回一个机器人对象
3  robot = Robot.RPC('192.168.58.2')
4  #恒力控制
5  status = 1 #恒力控制开启标志，0-关，1-开
6  sensor_num = 1 #力传感器编号
7  is_select = [0,0,1,0,0,0] #六个自由度选择[fx,fy,fz,mx,my,mz]，0-不生效，1-生效
8  force_torque = [0.0,0.0,-10.0,0.0,0.0,0.0]
9  gain = [0.0005,0.0,0.0,0.0,0.0,0.0] #力PID参数，力矩PID参数
10 adj_sign = 0 #自适应启停状态，0-关闭，1-开启
11 ILC_sign = 0 #ILC控制启停状态，0-停止，1-训练，2-实操
12 max_dis = 100.0 #最大调整距离
13 max_ang = 0.0 #最大调整角度
14 J1=[70.395, -46.976, 90.712, -133.442, -87.076, -27.138]
15 P2=[-123.978, -674.129, 44.308, -178.921, 2.734, -172.449]
16 P3=[123.978, -674.129, 42.308, -178.921, 2.734, -172.449]
17 error = robot.MoveJ(J1,1,0)
18 print("关节空间运动指令错误码",error)
19 error = robot.MoveL(P2,1,0)
20 print("笛卡尔空间直线运动指令错误码",error)
21 error = robot.FT_Control(status,sensor_num,is_select,force_torque,gain,adj_sign,
ILC_sign,max_dis,max_ang)
22 print("恒力控制开启错误码",error)
23 error = robot.MoveL(P3,1,0) #笛卡尔空间直线运动
24 print("笛卡尔空间直线运动指令错误码",error)
25 status = 0
26 error = robot.FT_Control(status,sensor_num,is_select,force_torque,gain,adj_sign,
ILC_sign,max_dis,max_ang)
27 print("恒力控制结束错误码",error)
```

10.14. 螺旋线探索

原型	<code>FT_SpiralSearch(rcs, ft, dr=0.7, max_t_ms=60000, max_vel=5)</code>
----	--

描述	螺旋线探索
必选参数	<ul style="list-style-type: none"><code>r_{cs}</code>：参考坐标系，0-工具坐标系，1-基坐标系<code>ft</code>：力或力矩阈值 (0~100)，单位 N 或 Nm;
默认参数	<ul style="list-style-type: none"><code>dr</code>：每圈半径进给量，单位 mm 默认0.7;<code>max_{t_ms}</code>：最大探索时间，单位 ms 默认 60000;<code>max_{vel}</code>：线速度最大值，单位 mm/s 默认 5
返回值	错误码 成功-0 失败- errcode

10.14.1. 代码示例

```
1 from fairino import Robot
2 # 与机器人控制器建立连接, 连接成功返回一个机器人对象
3 robot = Robot.RPC('192.168.58.2')
4 P = [36.794,-675.119, 65.379, -176.938, 2.535, -179.829]
5 #恒力参数
6 status = 1 #恒力控制开启标志, 0-关, 1-开
7 sensor_num = 1 #力传感器编号
8 is_select = [0,0,1,0,0,0] #六个自由度选择[fx,fy,fz,mx,my,mz], 0-不生效, 1-生效
9 force_torque = [0.0,0.0,-10.0,0.0,0.0,0.0]
10 gain = [0.0001,0.0,0.0,0.0,0.0,0.0] #力PID参数, 力矩PID参数
11 adj_sign = 0 #自适应启停状态, 0-关闭, 1-开启
12 ILC_sign = 0 #ILC控制启停状态, 0-停止, 1-训练, 2-实操
13 max_dis = 100.0 #最大调整距离
14 max_ang = 5.0 #最大调整角度
15 #螺旋线探索参数
16 rcs = 0 #参考坐标系, 0-工具坐标系, 1-基坐标系
17 fFinish = 10 #力或力矩阈值 (0~100), 单位N或Nm
18 error = robot.MoveL(P,1,0) #笛卡尔空间直线运动至初始点
19 print("笛卡尔空间直线运动至初始点",error)
20 is_select = [0,0,1,1,1,0] #六个自由度选择[fx,fy,fz,mx,my,mz], 0-不生效, 1-生效
21 error = robot.FT_Control(status,sensor_num,is_select,force_torque,gain,adj_sign,
ILC_sign, max_dis,max_ang)
22 print("恒力控制开启错误码",error)
23 error = robot.FT_SpiralSearch(rcs,fFinish,max_vel=3)
24 print("螺旋线探索错误码",error)
25 status = 0
26 error = robot.FT_Control(status,sensor_num,is_select,force_torque,gain,adj_sign,
ILC_sign, max_dis,max_ang)
27 print("恒力控制关闭错误码",error)
```

10.15. 旋转插入

原型	<code>FT_RotInsertion(r_{cs}, ft, or_n, angVelRot=3, angleMax=45, angAccmax=0, ro_{rn}=1)</code>
描述	旋转插入
必选参数	<ul style="list-style-type: none"><code>r_{cs}</code>：参考坐标系，0-工具坐标系，1-基坐标系；<code>ft</code>：力或力矩阈值 (0~100)，单位 N 或 Nm;<code>or_n</code>：力/扭矩方向，1-沿z轴方向，2-绕z轴方向;

默认参数	<ul style="list-style-type: none"><code>angVelRot</code>：旋转角速度，单位$^{\circ}/s$,默认 3;<code>angleMax</code>：最大旋转角度，单位$^{\circ}$ 默认 45;<code>angAccmax</code>：最大旋转加速度，单位$^{\circ}/s^2$，暂不使用 默认0;<code>rotorn</code>：旋转方向，1-顺时针，2-逆时针 默认1
返回值	错误码 成功-0 失败- errcode

10.15.1. 代码示例

```
1 from fairino import Robot
2 # 与机器人控制器建立连接，连接成功返回一个机器人对象
3 robot = Robot.RPC('192.168.58.2')
4 P = [36.794, -675.119, 65.379, -176.938, 2.535, -179.829]
5 #恒力参数
6 status = 1 #恒力控制开启标志，0-关，1-开
7 sensor_num = 1 #力传感器编号
8 is_select = [0,0,1,0,0,0] #六个自由度选择[fx,fy,fz,mx,my,mz]，0-不生效，1-生效
9 force_torque = [0.0,0.0,-10.0,0.0,0.0,0.0]
10 gain = [0.0001,0.0,0.0,0.0,0.0,0.0] #力PID参数，力矩PID参数
11 adj_sign = 0 #自适应启停状态，0-关闭，1-开启
12 ILC_sign = 0 #ILC控制启停状态，0-停止，1-训练，2-实操
13 max_dis = 100.0 #最大调整距离
14 max_ang = 5.0 #最大调整角度
15 #旋转插入参数
16 rcs = 0 #参考坐标系，0-工具坐标系，1-基坐标系
17 forceInsertion = 2.0 #力或力矩阈值 (0~100)，单位N或Nm
18 orn = 1 #力的方向，1-fz,2-mz
19 #默认参数 angVelRot：旋转角速度，单位 $^{\circ}/s$  默认 3
20 #默认参数 angleMax：最大旋转角度，单位 $^{\circ}$  默认 5
21 #默认参数 angAccmax：最大旋转加速度，单位 $^{\circ}/s^2$ ，暂不使用 默认0
22 #默认参数 rotorn：旋转方向，1-顺时针，2-逆时针 默认1
23 error = robot.MoveL(P,1,0) #笛卡尔空间直线运动至初始点
24 print("笛卡尔空间直线运动至初始点",error)
25 error = robot.FT_Control(status,sensor_num,is_select,force_torque,gain,adj_sign,
ILC_sign,max_dis,max_ang)
26 print("恒力控制开启错误码",error)
27 error = robot.FT_RotInsertion(rcs,1,orn)
28 print("旋转插入错误码",error)
29 status = 0
30 error = robot.FT_Control(status,sensor_num,is_select,force_torque,gain,adj_sign,
ILC_sign,max_dis,max_ang)
31 print("恒力控制关闭错误码",error)
```

10.16. 直线插入

原型	<code>FT_LinInsertion(rcs, ft, disMax, linorn, lin_v=1.0, lin_a=1.0)</code>
描述	直线插入
必选参数	<ul style="list-style-type: none"><code>rcs</code>：参考坐标系，0-工具坐标系，1-基坐标系<code>ft</code>：力或力矩阈值 (0~100)，单位 N 或 Nm;<code>disMax</code>：最大插入距离，单位 mm;<code>linorn</code>：插入方向:0-负方向，1-正方向

默认参数	<ul style="list-style-type: none"> <code>lin_v</code>: 直线速度, 单位 mm/s 默认1; <code>lin_a</code>: 直线加速度, 单位 mm/s², 暂不使用 默认1
返回值	错误码 成功-0 失败- errcode

10.16.1. 代码示例

```

1  from fairino import Robot
2  # 与机器人控制器建立连接, 连接成功返回一个机器人对象
3  robot = Robot.RPC('192.168.58.2')
4  P = [36.794, -675.119, 65.379, -176.938, 2.535, -179.829]
5  #恒力参数
6  status = 1 #恒力控制开启标志, 0-关, 1-开
7  sensor_num = 1 #力传感器编号
8  is_select = [0,0,1,0,0,0] #六个自由度选择[fx,fy,fz,mx,my,mz], 0-不生效, 1-生效
9  force_torque = [0.0,0.0,-10.0,0.0,0.0,0.0]
10 gain = [0.0001,0.0,0.0,0.0,0.0,0.0] #力PID参数, 力矩PID参数
11 adj_sign = 0 #自适应启停状态, 0-关闭, 1-开启
12 ILC_sign = 0 #ILC控制启停状态, 0-停止, 1-训练, 2-实操
13 max_dis = 100.0 #最大调整距离
14 max_ang = 5.0 #最大调整角度
15 #直线插入参数
16 rcs = 0 #参考坐标系, 0-工具坐标系, 1-基坐标系
17 force_goal = 10.0 #力或力矩阈值 (0~100), 单位N或Nm
18 disMax = 100.0 #最大插入距离, 单位mm
19 linorn = 1 #插入方向, 1-正方向, 2-负方向
20 #默认参数 lin_v: 直线速度, 单位 mm/s 默认1
21 #默认参数 lin_a: 直线加速度, 单位 mm/s^2, 暂不使用 默认0
22 error = robot.MoveL(P,1,0) #笛卡尔空间直线运动至初始点
23 print("笛卡尔空间直线运动至初始点",error)
24 error = robot.FT_Control(status,sensor_num,is_select,force_torque,gain,adj_sign,
ILC_sign,max_dis,max_ang)
25 print("恒力控制开启错误码",error)
26 error = robot.FT_LinInsertion(rcs,force_goal,disMax,linorn)
27 print("直线插入错误码",error)
28 status = 0
29 error = robot.FT_Control(status,sensor_num,is_select,force_torque,gain,adj_sign,
ILC_sign,max_dis,max_ang)
30 print("恒力控制关闭错误码",error)

```

10.17. 计算中间平面位置开始

原型	<code>FT_CalCenterStart()</code>
描述	计算中间平面位置开始
必选参数	无
默认参数	无
返回值	错误码 成功-0 失败- errcode



10.18. 计算中间平面位置结束

原型	<code>FT_CalCenterEnd()</code>
描述	计算中间平面位置结束
必选参数	无
默认参数	无
返回值	错误码 成功-0 失败- errcode - <code>pos=[x,y,z,rx,ry,rz]</code> ：中间平面位置

10.19. 表面定位

原型	<code>FT_FindSurface (rcs, dir, axis, disMax, ft, lin_v=3.0, lin_a=0.0)</code>
描述	表面定位
必选参数	<ul style="list-style-type: none"><code>rcs</code>：参考坐标系，0-工具坐标系，1-基坐标系；<code>dir</code>：移动方向，1-正方向，2-负方向；<code>axis</code>：移动轴，1-x，2-y，3-z；<code>disMax</code>：最大探索距离，单位 mm；<code>ft</code>：动作终止力阈值，单位N；
默认参数	<ul style="list-style-type: none"><code>lin_v</code>：探索直线速度，单位mm/s 默认3；<code>lin_a</code>：探索直线加速度，单位mm/s^2 默认0；
返回值	错误码 成功-0 失败- errcode

10.19.1. 代码示例

```
1  from fairino import Robot
2  import time
3  # 与机器人控制器建立连接, 连接成功返回一个机器人对象
4  robot = Robot.RPC('192.168.58.2')
5  #恒力控制
6  status = 1 #恒力控制开启标志, 0-关, 1-开
7  sensor_num = 1 #力传感器编号
8  is_select = [1,0,0,0,0,0] #六个自由度选择[fx,fy,fz,mx,my,mz], 0-不生效, 1-生效
9  force_torque = [-2.0,0.0,0.0,0.0,0.0,0.0]
10 gain = [0.0002,0.0,0.0,0.0,0.0,0.0] #力PID参数, 力矩PID参数
11 adj_sign = 0 #自适应启停状态, 0-关闭, 1-开启
12 ILC_sign = 0 #ILC控制启停状态, 0-停止, 1-训练, 2-实操
13 max_dis = 15.0 #最大调整距离
14 max_ang = 0.0 #最大调整角度
15 #表面定位参数
16 rcs = 0 #参考坐标系, 0-工具坐标系, 1-基坐标系
17 direction = 1 #移动方向, 1-正方向, 2-负方向
18 axis = 1 #移动轴, 1-X,2-Y,3-Z
19 lin_v = 3.0 #探索直线速度, 单位mm/s
20 lin_a = 0.0 #探索直线加速度, 单位mm/s^2
21 disMax = 50.0 #最大探索距离, 单位mm
22 force_goal = 2.0 #动作终止力阈值, 单位N
23 P1=[-77.24,-679.599,58.328,179.373,-0.028,-167.849]
24 Robot.MoveCart(P1,1,0) #关节空间点到点运动
25 #x方向寻找中心
26 #第1个表面
27 error = robot.FT_CalCenterStart()
28 print("计算中间平面开始错误码",error)
29 error = robot.FT_Control(status,sensor_num,is_select,force_torque,gain,adj_sign,
ILC_sign,max_dis,max_ang)
30 print("恒力控制开始错误码",error)
31 error = robot.FT_FindSurface(rcs,direction,axis,disMax,force_goal)
32 print("寻找X+表面错误码",error)
33 status = 0
34 error = robot.FT_Control(status,sensor_num,is_select,force_torque,gain,adj_sign,
ILC_sign,max_dis,max_ang)
35 print("恒力控制结束错误码",error)
36 time.sleep(2)
37 error = robot.MoveCart(P1,1,0) #关节空间点到点运动
38 print("关节空间点到点运动错误码",error)
39 time.sleep(5)
40 #第2个表面
41 error = robot.FT_Control(status,sensor_num,is_select,force_torque,gain,adj_sign,
ILC_sign,max_dis,max_ang)
42 print("恒力控制开始错误码",error)
43 direction = 2 #移动方向, 1-正方向, 2-负方向
44 error = robot.FT_FindSurface(rcs,direction,axis,disMax,force_goal)
45 print("寻找X-表面错误码",error)
46 status = 0
47 error = robot.FT_Control(status,sensor_num,is_select,force_torque,gain,adj_sign,
ILC_sign,max_dis,max_ang)
48 print("恒力控制结束错误码",error)
49 #计算x方向中心位置
50 error,xcenter = robot.FT_CalCenterEnd()
51 print("计算X方向中间平面结束错误码",xcenter)
52 error = robot.MoveCart(xcenter,1,0)
53 print("关节空间点到点运动错误码",error)
54 time.sleep(1)
55 #y方向寻找中心
56 #第1个表面
```

```
57 error =robot.FT_CalCenterStart()
58 print("计算中间平面开始错误码",error)
59 error =robot.FT_Control(status,sensor_num,is_select,force_torque,gain,adj_sign,
ILC_sign,max_dis,max_ang)
60 print("恒力控制开始错误码",error)
61 direction = 1 #移动方向, 1-正方向, 2-负方向
62 axis = 2 #移动轴, 1-X, 2-Y, 3-Z
63 disMax = 150.0 #最大探索距离, 单位mm
64 lin_v = 6.0 #探索直线速度, 单位mm/s
65 error =robot.FT_FindSurface(rcs,direction,axis,disMax,force_goal)
66 print("寻找表面Y+错误码",error)
67 status = 0
68 error =robot.FT_Control(status,sensor_num,is_select,force_torque,gain,adj_sign,
ILC_sign,max_dis,max_ang)
69 print("恒力控制结束错误码",error)
70 error =robot.MoveCart(P1,1,0) #关节空间点到点运动
71 print("关节空间点到点运动错误码",error)
72 Robot.WaitMs(1000)
73 #第2个表面
74 error =robot.FT_Control(status,sensor_num,is_select,force_torque,gain,adj_sign,
ILC_sign,max_dis,max_ang)
75 print("恒力控制开始错误码",error)
76 direction = 2 #移动方向, 1-正方向, 2-负方向
77 error =robot.FT_FindSurface(rcs,direction,axis,disMax,force_goal)
78 print("寻找表面Y-错误码",error)
79 status = 0
80 error =robot.FT_Control(status,sensor_num,is_select,force_torque,gain,adj_sign,
ILC_sign,max_dis,max_ang)
81 print("恒力控制结束错误码",error)
82 #计算y方向中心位置
83 error,ycenter=robot.FT_CalCenterEnd()
84 print("计算中间平面Y方向结束错误码",ycenter)
85 error =robot.MoveCart(ycenter,1,0)
86 print("关节空间点到点运动错误码",error)
```

10.20. 柔顺控制关闭

原型	<code>FT_ComplianceStop()</code>
描述	柔顺控制关闭
必选参数	无
默认参数	无
返回值	错误码 成功-0 失败- errcode

10.21. 柔顺控制开启

原型	<code>FT_ComplianceStart(p, force)</code>
描述	柔顺控制开启
必选参数	<ul style="list-style-type: none"><code>p</code>: 位置调节系数或柔顺系数<code>force</code>: 柔顺开启力阈值, 单位N
默认参数	无



latest



10.21.1. 代码示例

```

1  from fairino import Robot
2  import time
3  # 与机器人控制器建立连接, 连接成功返回一个机器人对象
4  robot = Robot.RPC('192.168.58.2')
5  J1=[75.005,-46.434,90.687,-133.708,-90.315,-27.139]
6  P2=[-77.24,-679.599,38.328,179.373,-0.028,-167.849]
7  P3=[77.24,-679.599,38.328,179.373,-0.028,-167.849]
8  #恒力控制参数
9  status = 1 #恒力控制开启标志, 0-关, 1-开
10 sensor_num = 1 #力传感器编号
11 is_select = [1,1,1,0,0,0] #六个自由度选择[fx,fy,fz,mx,my,mz], 0-不生效, 1-生效
12 force_torque = [-10.0,-10.0,-10.0,0.0,0.0,0.0]
13 gain = [0.0005,0.0,0.0,0.0,0.0,0.0] #力PID参数, 力矩PID参数
14 adj_sign = 0 #自适应启停状态, 0-关闭, 1-开启
15 ILC_sign = 0 #ILC控制启停状态, 0-停止, 1-训练, 2-实操
16 max_dis = 1000.0 #最大调整距离
17 max_ang = 0.0 #最大调整角度
18 error = robot.MoveJ(J1,1,0)
19 print("关节空间运动到点1错误码",error)
20 #柔顺控制
21 error = robot.FT_Control(status,sensor_num,is_select,force_torque,gain,adj_sign,
ILC_sign,max_dis,max_ang)
22 print("恒力控制开始错误码",error)
23 p = 0.00005 #位置调节系数或柔顺系数
24 force = 30.0 #柔顺开启力阈值, 单位N
25 error = robot.FT_ComplianceStart(p,force)
26 print("柔顺控制开始错误码",error)
27 error = robot.MoveL(P2,1,0,vel =10) #笛卡尔空间直线运动
28 print("笛卡尔空间直线运动到点2错误码", error)
29 error = robot.MoveL(P3,1,0,vel =10)
30 print("笛卡尔空间直线运动到点3错误码", error)
31 time.sleep(1)
32 error = robot.FT_ComplianceStop()
33 print("柔顺控制结束错误码",error)
34 status = 0
35 error = robot.FT_Control(status,sensor_num,is_select,force_torque,gain,adj_sign,
ILC_sign,max_dis,max_ang)
36 print("恒力控制关闭错误码",error)

```

10.22. 负载辨识滤波初始化

在 python 版本加入: SDK-v2.0.1

原型	<code>LoadIdentifyDynFilterInit()</code>
描述	负载辨识滤波初始化
必选参数	无
默认参数	无
返回值	错误码 成功-0 失败- errcode



latest



10.22.1. 代码示例

```
1  from fairino import Robot
2
3  # 与机器人控制器建立连接, 连接成功返回一个机器人对象
4  robot = Robot.RPC('192.168.58.2')
5
6  #负载辨识滤波初始化
7  error = robot.LoadIdentifyDynFilterInit()
8  print("LoadIdentifyDynFilterInit:",error)
9  #负载辨识变量初始化
10 error = robot.LoadIdentifyDynVarInit()
11 print("LoadIdentifyDynVarInit:",error)
12
13 joint_torque= [0,0,0,0,0,0]
14 joint_pos= [0,0,0,0,0,0]
15 gain=[0,0.05,0,0,0,0,0,0.02,0,0,0,0]
16 t =10
17 error,joint_pos=robot.GetActualJointPosDegree(1)
18 joint_pos[1] = joint_pos[1]+10
19 error,joint_torque=robot.GetJointTorques(1)
20 joint_torque[1] = joint_torque[1]+2
21 #负载辨识主程序
22 error = robot.LoadIdentifyMain(joint_torque, joint_pos, t)
23 print("LoadIdentifyMain:",error)
24 #获取负载辨识结果
25 error = robot.LoadIdentifyGetResult(gain)
26 print("LoadIdentifyGetResult:",error)
```

10.23. 负载辨识变量初始化

在 python 版本加入: SDK-v2.0.1

原型	<code>LoadIdentifyDynVarInit()</code>
描述	负载辨识变量初始化
必选参数	无
默认参数	无
返回值	错误码 成功-0 失败- errcode

10.24. 负载辨识主程序

在 python 版本加入: SDK-v2.0.1

原型	<code>LoadIdentifyMain(joint_torque, joint_pos, t)</code>
描述	负载辨识主程序

必选参数	<ul style="list-style-type: none"><code>joint_torque</code>：关节扭矩 j1-j6；<code>joint_pos</code>：关节位置 j1-j6<code>t</code>：采样周期
默认参数	无
返回值	错误码 成功-0 失败- errcode

10.25. 获取负载辨识结果

在 python 版本加入: SDK-v2.0.1

原型	<code>LoadIdentifyGetResult(gain)</code>
描述	获取负载辨识结果
必选参数	<ul style="list-style-type: none"><code>gain</code>：重力项系数double[6]，离心项系数double[6]
默认参数	无
返回值	<ul style="list-style-type: none">错误码 成功-0 失败- errcode<code>weight</code>：负载重量<code>cog=[x,y,z]</code>：负载质心坐标

10.26. 力传感器辅助拖动

在 python 版本加入: SDK-v2.0.5

原型	<code>ForceAndJointImpedanceStartStop(status, impedanceFlag, lamdeDain, KGain, BGain, dragMaxTcpVel, dragMaxTcpOriVel)</code>
描述	力传感器辅助拖动
必选参数	<ul style="list-style-type: none"><code>status</code>：控制状态，0-关闭；1-开启<code>impedanceFlag</code>：阻抗开启标志，0-关闭；1-开启<code>lamdeDain</code>：[D1,D2,D3,D4,D5, D6] 拖动增益<code>KGain</code>：[K1,K2,K3,K4,K5,K6]刚度增益<code>BGain</code>：[B1,B2,B3,B4,B5,B]阻尼增益<code>dragMaxTcpVel</code>：拖动末端最大线速度限制<code>dragMaxTcpOriVel</code>：拖动末端最大角速度限制
默认参数	无
返回值	错误码 成功-0 失败- errcode

10.26.1. 代码示例

```
1  from fairino import Robot
2  import time
3  # 与机器人控制器建立连接, 连接成功返回一个机器人对象
4
5  robot = Robot.RPC('192.168.58.2')
6
7  status = 1 #控制状态, 0-关闭; 1-开启
8  asaptiveFlag = 1 #自适应开启标志, 0-关闭; 1-开启
9  interfereDragFlag = 1 #干涉区拖动标志, 0-关闭; 1-开启
10 ingularityConstraintsFlag = 0 #奇异点策略, 0-规避; 1-穿越
11 M = [15, 15, 15, 0.5, 0.5, 0.1] #惯性系数
12 B = [150, 150, 150, 5, 5, 1] #阻尼系数
13 K = [0, 0, 0, 0, 0, 0] #刚度系数
14 F = [5, 5, 5, 1, 1, 1] #拖动六维力阈值
15 Fmax = 50 #最大拖动力限制
16 Vmax = 1810 #最大关节速度限制
17
18 error = robot.EndForceDragControl(status, asaptiveFlag, interfereDragFlag,
ingularityConstraintsFlag, M, B, K, F, Fmax, Vmax)
19 print("EndForceDragControl return:",error)
20
21 time.sleep(10)
22 status=0
23 error = robot.EndForceDragControl(status, asaptiveFlag, interfereDragFlag,
ingularityConstraintsFlag, M, B, K, F, Fmax, Vmax)
24 print("EndForceDragControl return:",error)
```

10.27. 报错清除后力传感器自动开启

在 python 版本加入: SDK-v2.0.5

原型	<code>SetForceSensorDragAutoFlag(status)</code>
描述	报错清除后力传感器自动开启
必选参数	<ul style="list-style-type: none"><code>status</code>: 控制状态, 0-关闭; 1-开启
默认参数	无
返回值	错误码 成功-0 失败- errcode

10.27.1. 代码示例

```
1  from fairino import Robot
2  import time
3  # 与机器人控制器建立连接, 连接成功返回一个机器人对象
4  robot = Robot.RPC('192.168.58.2')
5
6  error = robot.SetForceSensorDragAutoFlag (1)
7  print("SetForceSensorDragAutoFlag return:",error)
```



latest



10.28. 设置六维力和关节阻抗混合拖动开关及参数

在 python 版本加入: SDK-v2.0.5

原型	<code>EndForceDragControl(status, asaptiveFlag, interfereDragFlag, ingularityConstraintsFlag, M, B, K, F, Fmax, Vmax)</code>
描述	设置六维力和关节阻抗混合拖动开关及参数
必选参数	<ul style="list-style-type: none"><code>status</code> : 控制状态, 0-关闭; 1-开启<code>asaptiveFlag</code> : 自适应开启标志, 0-关闭; 1-开启<code>interfereDragFlag</code> : 干涉区拖动标志, 0-关闭; 1-开启<code>ingularityConstraintsFlag</code> : 奇异点策略, 0-规避; 1-穿越<code>M=[m1,m2,m3,m4,m5,m6]</code> : 惯性系数<code>B=[b1,b2,b3,b4,b5,b6]</code> : 阻尼系数<code>K=[k1,k2,k3,k4,k5,k6]</code> : 刚度系数<code>F=[f1,f2,f3,f4,f5,f6]</code> : 拖动六维力阈值<code>Fmax</code> : 最大拖动力限制<code>Vmax</code> : 最大关节速度限制
默认参数	无
返回值	错误码 成功-0 失败- errcode

10.28.1. 代码示例

```
1  from fairino import Robot
2  import time
3  # 与机器人控制器建立连接, 连接成功返回一个机器人对象
4
5  robot = Robot.RPC('192.168.58.2')
6
7  status = 1 #控制状态, 0-关闭; 1-开启
8  impedanceFlag = 1 #阻抗开启标志, 0-关闭; 1-开启
9  lamdeDain = [ 3.0, 2.0, 2.0, 2.0, 2.0, 3.0] # 拖动增益
10 KGain = [0.00, 0.00, 0.00, 0.00, 0.00, 0.00] # 刚度增益
11 BGain = [150, 150, 150, 5.0, 5.0, 1.0] # 阻尼增益
12 dragMaxTcpVel = 1000 #拖动末端最大线速度限制
13 dragMaxTcpOriVel = 180 #拖动末端最大角速度限制
14
15 error = robot.DragTeachSwitch(1)
16 print("DragTeachSwitch 1 return:",error)
17
18 error = robot.ForceAndJointImpedanceStartStop(status, impedanceFlag, lamdeDain,
KGain, BGain,dragMaxTcpVel,dragMaxTcpOriVel)
19 print("ForceAndJointImpedanceStartStop return:",error)
20
21 error = robot.GetForceAndTorqueDragState()
22 print("GetForceAndTorqueDragState return:",error)
23
24 time.sleep(10)
25
26 status = 0 #控制状态, 0-关闭; 1-开启
27 impedanceFlag = 0 #阻抗开启标志, 0-关闭; 1-开启
28 error = robot.ForceAndJointImpedanceStartStop(status, impedanceFlag, lamdeDain,
KGain, BGain,dragMaxTcpVel,dragMaxTcpOriVel)
29 print("ForceAndJointImpedanceStartStop return:",error)
30
31 error = robot.GetForceAndTorqueDragState()
32 print("GetForceAndTorqueDragState return:",error)
33
34 error = robot.DragTeachSwitch(0)
35 print("DragTeachSwitch 0 return:",error)
```

10.29. 获取力传感器拖动开关状态

在 python 版本加入: SDK-v2.0.5

原型	GetForceAndTorqueDragState()
描述	获取力传感器拖动开关状态
必选参数	无
默认参数	无
返回值	<ul style="list-style-type: none">• 错误码 成功-0 失败- errcode• dragState : 力传感器辅助拖动控制状态, 0-关闭; 1-开启• sixDimensionalDragState : 六维力辅助拖动控制状态, 0-关闭; 1-开启

10.30. 设置力传感器下负载重量

在 python 版本加入: SDK-v2.0.5

原型	SetForceSensorPayload(weight)
描述	设置力传感器下负载重量
必选参数	<ul style="list-style-type: none">weight: 负载重量 kg
默认参数	无
返回值	错误码 成功-0 失败- errcode

10.30.1. 代码示例

```
1  from fairino import Robot
2  import time
3  # 与机器人控制器建立连接, 连接成功返回一个机器人对象
4
5  robot = Robot.RPC('192.168.58.2')
6
7  error = robot.SetForceSensorPayload(0.8)
8  print("SetForceSensorPayload return:",error)
9
10 error = robot.SetForceSensorPayloadCog(0.5,0.6,12.5)
11 print("SetForceSensorPayLoadCog return:",error)
12
13 error = robot.GetForceSensorPayload()
14 print("GetForceSensorPayLoad return:",error)
15
16 error = robot.GetForceSensorPayloadCog()
17 print("GetForceSensorPayLoadCog return:",error)
```

10.31. 设置力传感器下负载质心

在 python 版本加入: SDK-v2.0.5

原型	SetForceSensorPayloadCog(x,y,z)
描述	设置力传感器下负载质心
必选参数	<ul style="list-style-type: none">x: 负载质心x mmy: 负载质心y mmz: 负载质心z mm
默认参数	无
返回值	错误码 成功-0 失败- errcode

10.32. 获取力传感器下负载重量

在 python 版本加入: SDK-v2.0.5

原型	<code>GetForceSensorPayload()</code>
描述	获取力传感器下负载重量
必选参数	无
默认参数	无
返回值	<ul style="list-style-type: none">错误码 成功-0 失败- errcode<code>weight</code>: 负载重量 kg

10.33. 获取力传感器下负载质心

在 python 版本加入: SDK-v2.0.5

原型	<code>GetForceSensorPayloadCog()</code>
描述	获取力传感器下负载质心
必选参数	无
默认参数	无
返回值	<ul style="list-style-type: none">错误码 成功-0 失败- errcode<code>x</code>: 负载质心x mm<code>y</code>: 负载质心y mm<code>z</code>: 负载质心z mm

10.34. 力传感器自动校零

在 python 版本加入: SDK-v2.0.5

原型	<code>ForceSensorAutoComputeLoad()</code>
描述	力传感器自动校零
必选参数	无
默认参数	无
返回值	<ul style="list-style-type: none">错误码 成功-0 失败- errcode<code>weight</code>: 传感器质量 kg<code>pos=[x,y,z]</code>: 传感器质心 mm

10.34.1. 代码示例

```
1  from fairino import Robot
2  # 与机器人控制器建立连接, 连接成功返回一个机器人对象
3
4  robot = Robot.RPC('192.168.58.2')
5
6  error = robot.SetForceSensorPayload(0)
7  print("SetForceSensorPayload return:",error)
8
9  error = robot.SetForceSensorPayloadCog(0,0,0)
10 print("SetForceSensorPayloadCog return:",error)
11
12 error = robot.ForceSensorAutoComputeLoad()
13 print("ForceSensorAutoComputeLoad return:",error)
```

10.35. 传感器自动校零数据记录

在 python 版本加入: SDK-v2.0.5

原型	<code>ForceSensorSetSaveDataFlag(recordCount)</code>
描述	传感器自动校零数据记录
必选参数	<ul style="list-style-type: none"><code>recordCount</code>: 记录数据个数 1-3
默认参数	无
返回值	错误码 成功-0 失败-errcode

10.36. 传感器自动校零计算

在 python 版本加入: SDK-v2.0.5

原型	<code>ForceSensorComputeLoad()</code>
描述	传感器自动校零数据记录
必选参数	无
默认参数	无
返回值	<ul style="list-style-type: none">错误码 成功-0 失败-errcode<code>weight</code>: 传感器质量 kg<code>pos=[x,y,z]</code>: 传感器质心 mm

