[伶](https://fairino-doc-zhs.readthedocs.io/latest/index.html) / [SDK](https://fairino-doc-zhs.readthedocs.io/latest/SDKManual/index.html) / [Python](https://fairino-doc-zhs.readthedocs.io/latest/SDKManual/python_intro.html) / 7. 机器⼈轨迹复现



**7.** 机器⼈轨迹复现

**7.1.** 设置轨迹记录参数

|  |
| --- |
| SetTPDParam(name, period\_ms, type=1,di\_choose=0, do\_choose=0) |

|  |
| --- |
| 设置轨迹记录参数 |
| name ：轨迹名；  period\_ms ：采样周期， 固定值，2ms 或 4ms 或 8ms; |
| type ：数据类型，1-关节位置；  di\_choose ：DI 选择,bit0~bit7 对应控制箱 DI0~DI7 ，bit8~bit9 对应末端DI0~DI1 ，0-不选择，1-选择  do\_choose ：DO 选择,bit0~bit7 对应控制箱 DO0~DO7 ，bit8~bit9 对应末端 DO0~DO1 ，0-不选择， |
| 错误码 成功-0 失败- errcode |

**7.1.1.** 代码示例

 latest 

|  |  |  |
| --- | --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24 | **from fairino import** Robot  **import time**  *# 与机器⼈控制器建⽴连接，* *连接成功返回—个机器⼈对象*  robot = Robot.RPC( I192.168.58.2I) type = 1 *# 数据类型，1-关节位置*  name = Itpd2023I *# 轨迹名*  period = 4 *#采样周期，2ms或4ms或8ms* di = 0 *# di输⼊配置*  do = 0 *# do输出配置*  ret = robot.SetTPDParam(name, period, di\_choose=di)  print("配置TPD参数错误码", ret)  robot.Mode(1) *# 机器⼈切⼊⼿动模式* time.sleep(1)  robot.DragTeachSwitch(1) *#机器⼈切⼊拖动示教模式* ret = robot.GetActualTCPPose()  print("获取当前⼯具位姿", ret)  time.sleep(1)  ret = robot.SetTPDStart(name, period, do\_choose=do) print("开始记录示教轨迹错误码", ret)  time.sleep(15)  ret = robot.SetWebTPDStop() *# 停⽌记录示教轨迹* print("停⽌记录示教轨迹错误码", ret)  robot.DragTeachSwitch(0) *#机器⼈切⼊⾮拖动示教模式*  *# robot.SetTPDDelete('tpd2023') # 删除TPD轨迹* | *#配置TPD参数*  *# 开始记录示教轨迹* |

**7.2.** 开始轨迹记录

|  |
| --- |
| SetTPDStart(name, period\_ms, type=1,di\_choose=0, do\_choose=0) |

开始轨迹记录

. name ：轨迹名；

. period\_ms ：采样周期， 固定值，2ms或4ms或8ms；

. type ：数数据类型，1-关节位置 默认1;

. di\_choose ：DI 选择,bit0~bit7 对应控制箱 DI0~DI7 ，bit8~bit9 对应末端DI0~DI1 ，0-不选择，1-选择

. do\_choose ：DO 选择,bit0~bit7 对应控制箱 DO0~DO7 ，bit8~bit9 对应末端 DO0~DO1 ，0-不选择，

错误码 成功-0 失败- errcode

**7.3.** 停⽌轨迹记录

|  |  |  |  |
| --- | --- | --- | --- |
| 原型 |  |  |  |
| SetWebTPDStop() |
|  | |
| 描述 | 停⽌轨迹记录 | | |
| 必选参数 | ⽆ | | |
| 默认参数 | ⽆ | | |
| 返回值 | 错误码 成功-0 失败- errcode | | |

**7.4.** 删除轨迹记录

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 原型 |  |  |  |  |  |
| SetTPDDelete(name) | | |
|  |  |  |
| 描述 | 删除轨迹记录 | | | | |
| 必选参数 | . name :轨迹名 | | | | |
| 默认参数 | ⽆ | | | | |
| 返回值 | 错误码 成功-0 失败- errcode | | | | |

**7.5.** 轨迹预加载

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 原型 |  |  |  |  |  |
| LoadTPD(name) | | |
|  | | | |
| 描述 | 轨迹预加载 | | | | |
| 必选参数 |  | |  | :轨迹名 | |
| name |
|  |
| 默认参数 | ⽆ | | | | |
| 返回值 | 错误码 成功-0 失败- errcode | | | | |



 latest 

**7.5.1.** 代码示例

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | **from fairino import** Robot  **import time**  *# 与机器⼈控制器建⽴连接，* *连接成功返回—个机器⼈对象*  robot = Robot.RPC( I192.168.58.2I)  *# P1=[-321.821, 125.694, 282.556, 174.106, -15.599, 152.669]*  name = Itpd2023I *#轨迹名*  blend = 1 *#是否平滑，1-平滑，0-不平滑* ovl = 100.0 *#速度缩放*  ret = robot.LoadTPD(name) *#轨迹预加载* print("轨迹预加载错误码",ret)  ret,P1 = robot.GetTPDStartPose(name) *#获取轨迹起始位姿* print ("获取轨迹起始位姿错误码",ret,"起始位姿",P1)  ret = robot.MoveL(P1,0,0) *#运动到起始点*  print("运动到起始点错误码",ret)  time.sleep(10)  ret = robot.MoveTPD(name, blend, ovl) *#轨迹复现* print("轨迹复现错误码",ret) |

**7.6.** 获取轨迹起始位姿

|  |  |  |  |
| --- | --- | --- | --- |
| 原型 | |  | | --- | | GetTPDStartPose(name) | | |
| 描述 | 获取轨迹起始位姿 | |
| 必选参数 | . name :轨迹名 | |
| 默认参数 | ⽆ | |
| 返回值 | 错误码 成功-0 失败- errcode | |
|  | |  | | --- | | desc\_pose= | |
| [x,y,z,rx,ry,rz] ：轨迹起始位姿 | |

**7.7.** 轨迹复现

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 原型 |  |  |  |  |  |  |  |
| MoveTPD(name,blend,ovl) | | | | |
|  | | | | | |
| 描述 | 轨迹复现 | | | | | | |
| 必选参数 |  | | name | | :轨迹名 | | |
|  |  |  | ：是否平滑，0-不平滑，1-平滑 速度缩放因⼦ ，范围[0~100] | |
| blend | | |
| ovl | : | |
| 默认参数 | ⽆ | | | | | | |
| 返回值 | 错误码 成功-0 失败- errcode | | | | | | |

**7.8.** 轨迹预处理

 latest

|  |  |  |
| --- | --- | --- |
| 原型 | |  | | --- | | LoadTrajectoryJ(name,ovl,opt=1) | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 描述 | 轨迹预处理 | | | |
| 必选参数 |  |  | | :轨迹名,如 ：/fruser/traj/trajHelix\_aima\_1.txt; |
| name | |
|  |  |
| ovl | ：速度缩放百分⽐ ，范围[0~100]; | |
| 默认参数 |  |  | ：1-控制点，默认为1 | |
| opt |
| 返回值 | 错误码 成功-0 失败- errcode | | | |

**7.9.** 轨迹复现

|  |  |  |  |
| --- | --- | --- | --- |
| 原型 |  |  |  |
| MoveTrajectoryJ() |
|  | |
| 描述 | 轨迹复现 | | |
| 必选参数 | ⽆ | | |
| 默认参数 | ⽆ | | |
| 返回值 | 错误码 成功-0 失败- errcode | | |

**7.10.** 获取轨迹起始位姿

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 原型 |  | |  | | --- | | GetTrajectoryStartPose(name) | | |
| 描述 | 获取轨迹起始位姿 | | |
| 必选参数 |  | name :轨迹名 | |
| 默认参数 | ⽆ | | |
| 返回值 | 错误码 成功-0 失败- errcode | | |
|  | | |  | | --- | | desc\_pose= | |
| [x,y,z,rx,ry,rz] ：轨迹起始位姿 | | |

**7.11.** 获取轨迹点编号

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 原型 |  |  |  |  |  |
| GetTrajectoryPointNum() | | |
|  |  |  |
| 描述 | 获取轨迹点编号 | | | | |
| 必选参数 | ⽆ | | | | |
| 默认参数 | ⽆ | | | | |
| 返回值 | 错误码 成功-0 失败- errcode . pnum ：轨迹点编号 | | | | |

**7.12.** 设置轨迹运⾏中的速度

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 原型 |  |  |  |  |
| SetTrajectoryJSpeed(ovl) | |
|  | | |
| 描述 | 设置轨迹运⾏中的速度 | | | |
| 必选参数 |  | ovl | :速度缩放百分⽐ ，范围[0~100] | |
|  | |
| 默认参数 | ⽆ | | | |
| 返回值 | 错误码 成功-0 失败- errcode | | | |

**7.13.** 设置轨迹运⾏中的⼒和扭矩

|  |  |  |  |
| --- | --- | --- | --- |
| 原型 | |  | | --- | | SetTrajectoryJForceTorque(ft) | |  |
| 描述 | 设置轨迹运⾏中的⼒和扭矩 | |
| 必选参数 | ft= [fx,fy,fz,tx,ty,tz] :单位N和Nm | |
| 默认参数 | ⽆ | |
| 返回值 | 错误码 成功-0 失败- errcode | |

**7.14.** 设置轨迹运⾏中的沿**x**⽅向的⼒

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 原型 |  |  |  |  |
| SetTrajectoryJForceFx(fx) | |
|  | | |
| 描述 | 设置轨迹运⾏中的沿x⽅向的⼒ | | | |
| 必选参数 | ft | | :沿x⽅向的⼒ ，单位N | |
| 默认参数 | ⽆ | | | |
| 返回值 | 错误码 成功-0 失败- errcode | | | |

**7.15.** 设置轨迹运⾏中的沿**y**⽅向的⼒

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 原型 |  |  |  |  |
| SetTrajectoryJForceFx(fy) | |
|  | | |
| 描述 | 设置轨迹运⾏中的沿y⽅向的⼒ | | | |
| 必选参数 | fy | | :沿y⽅向的⼒ ，单位N | |
| 默认参数 | ⽆ | | | |
| 返回值 | 错误码 成功-0 失败- errcode | | | |

**7.16.** 设置轨迹运⾏中的沿**z**⽅向的⼒

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 原型 |  |  |  |  |
| SetTrajectoryJForceFx(fz) | |
|  | | |
| 描述 | 设置轨迹运⾏中的沿z⽅向的⼒ | | | |
| 必选参数 | fz | | :沿z⽅向的⼒ ，单位N | |
| 默认参数 | ⽆ | | | |
| 返回值 | 错误码 成功-0 失败- errcode | | | |

**7.17.** 设置轨迹运⾏中的绕**x**轴的扭矩

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 原型 |  |  |  |  |
| SetTrajectoryJTorqueTx(tx) | |
|  | | |
| 描述 | 设置轨迹运⾏中的绕x轴的扭矩 | | | |
| 必选参数 | tx | | :绕x轴的扭矩，单位Nm | |
| 默认参数 | ⽆ | | | |
| 返回值 | 错误码 成功-0 失败- errcode | | | |

**7.18.** 设置轨迹运⾏中的绕**y**轴的扭矩

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 原型 |  |  |  |  |
| SetTrajectoryJTorqueTx(ty) | |
|  | | |
| 描述 | 设置轨迹运⾏中的绕y轴的扭矩 | | | |
| 必选参数 | ty | | :绕y轴的扭矩，单位Nm | |
| 默认参数 | ⽆ | | | |
| 返回值 | 错误码 成功-0 失败- errcode | | | |

**7.19.** 设置轨迹运⾏中的绕**z**轴的扭矩

*在* *Python 版本发⽣变更:* SDK-v2.0.8-3.7.8

 latest

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 原型 |  |  |  |  |  |
| SetTrajectoryJTorqueTx(tz) | | |
|  | | | |
| 描述 | 设置轨迹运⾏中的绕z轴的扭矩 | | | | |
| 必选参数 | . tz :绕z轴的扭矩，单位Nm | | | | |
| 默认参数 | ⽆ | | | | |
| 返回值 | 错误码 成功-0 失败- errcode | | | | |

**7.19.1.** 代码示例

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44 | **from fairino import** Robot  **import time**  *# 与机器⼈控制器建⽴连接，* *连接成功返回—个机器⼈对象*  robot = Robot.RPC( '192.168.58.2')  name = "/fruser/traj/trajHelix\_aima\_1.txt" *#轨迹名* blend = 1 *#是否平滑，1-平滑，0-不平滑*  ovl = 50.0 *#速度缩放*  ft = [0.5, 0.5, 0.5, 0.5, 0.5, 0.5]  ret = robot.LoadTrajectoryJ(name,ovl) *#轨迹预加载* print("轨迹预加载错误码",ret)  ret,P1 = robot.GetTrajectoryStartPose(name) *#获取轨迹起始位姿* print ("获取轨迹起始位姿错误码",ret,"起始位姿",P1)  ret = robot.MoveL(P1,1,0) *#运动到起始点*  print("运动到起始点错误码",ret)  ret = robot.GetTrajectoryPointNum() *#获取轨迹点编号*  print("获取轨迹点编号错误码",ret)  time.sleep(10)  ret = robot.MoveTrajectoryJ() *#轨迹复现* print("轨迹复现错误码",ret)  time.sleep(10)  ret = robot.SetTrajectoryJSpeed(ovl) *#设置轨迹运⾏中的速度* print("设置轨迹运⾏中的速度错误码",ret)  time.sleep(1)  ret = robot.SetTrajectoryJForceTorque(ft) *#设置轨迹运⾏中的⼒和扭矩* print("设置轨迹运⾏中的⼒和扭矩错误码",ret)  time.sleep(1)  ret = robot.SetTrajectoryJForceFx(0) *#设置轨迹运⾏中的沿x ⽅向的⼒* print("设置轨迹运⾏中的沿x⽅向的⼒错误码",ret)  time.sleep(1)  ret = robot.SetTrajectoryJForceFy(0) *#设置轨迹运⾏中的沿y⽅向的⼒* print("设置轨迹运⾏中的沿y⽅向的⼒错误码",ret)  time.sleep(1)  ret = robot.SetTrajectoryJForceFz(0) *#设置轨迹运⾏中的沿z ⽅向的⼒* print("设置轨迹运⾏中的沿z⽅向的⼒错误码",ret)  time.sleep(1)  ret = robot.SetTrajectoryJTorqueTx(0) *#设置轨迹运⾏中的绕x轴的扭矩* print("设置轨迹运⾏中的绕x轴的扭矩错误码",ret)  time.sleep(1)  ret = robot.SetTrajectoryJTorqueTy(0) *#设置轨迹运⾏中的绕y轴的扭矩* print("设置轨迹运⾏中的绕y轴的扭矩错误码",ret)  time.sleep(1)  ret = robot.SetTrajectoryJTorqueTz(0) *#设置轨迹运⾏中的绕z轴的扭矩* print("设置轨迹运⾏中的绕z轴的扭矩错误码",ret)  time.sleep(1) |

**7.20.** 上传轨迹**J**⽂件

*在* *Python 版本加⼊:* SDK-v2.0.8-3.7.8

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 原型 |  |  |  |  |
| TrajectoryJUpLoad(filePath) | |
|  |  |
| 描述 | 上传轨迹J⽂件 | | | |
| 必选参数 |  | | filePath :上传轨迹⽂件的全路径名，C://test/testJ.txt  latest | |

|  |  |
| --- | --- |
| 默认参数 | ⽆ |
| 返回值 | 错误码 成功-0 失败- errcode |

**7.21.** 删除轨迹**J**⽂件

*在* *Python 版本加⼊:* SDK-v2.0.8-3.7.8

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 原型 |  |  |  |  |  |
| TrajectoryJDelete(filePath) | | |
|  |  |  |
| 描述 | 删除轨迹J⽂件 | | | | |
| 必选参数 | . filePath :删除轨迹⽂件的全路径名，C://test/testJ.txt | | | | |
| 默认参数 | ⽆ | | | | |
| 返回值 | 错误码 成功-0 失败- errcode | | | | |

**7.21.1.** 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—个机器⼈对象* 4  robot = Robot.RPC( '192.168.58.2')  5  robot.LoggerInit()  6  robot.SetLoggerLevel(lvl=1)  7  8  retval = robot.TrajectoryJDelete("testA.txt")  9  print("TrajectoryJDelete return ", retval)  10  robot.TrajectoryJUpLoad("D://zUP/testA.txt")  11  12  traj\_file\_name = "/fruser/traj/testA.txt"  13  retval = robot.LoadTrajectoryJ(traj\_file\_name, 100, 1)  14  print("LoadTrajectoryJ return ", retval)  15  16  retval,traj\_start\_pose = robot.GetTrajectoryStartPose(traj\_file\_name)  17  print("GetTrajectoryStartPose return ", retval)  18  print("轨迹起始位姿:", traj\_start\_pose [0], traj\_start\_pose [1], traj\_start\_pose [2], traj\_start\_pose [3], traj\_start\_pose [4], traj\_start\_pose [5])  19  20  robot.SetSpeed(20)  21  robot.MoveCart(traj\_start\_pose, 1, 0)  22  23  time.sleep(5)  24  25  retval,traj\_num = robot.GetTrajectoryPointNum()  26  print("GetTrajectoryPointNum return ", retval)  27  print("轨迹点编号 : ", traj\_num)  28  29  retval = robot.MoveTrajectoryJ()  30  print("MoveTrajectoryJ return ", retval) |

**7.22.** 轨迹预处理**(**轨迹前瞻**)**

*在* *python 版本加⼊:* SDK-v2.1.0

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 原型 |  |  |  |  |  |  |
| LoadTrajectoryLA(name, mode, errorLim, type, precision, vamx, amax, jmax) | | | |
|  |  |  |  |
| 描述 | 轨迹预处理(轨迹前瞻) | | | | | |
| 必选参数 |  | | name :轨迹⽂件名  mode ：采样模式，0-不进⾏采样；1-等数据间隔采样；2-等误差限制采样  errorLim :误差限制，使⽤直线拟合⽣效 | | | |
| type :平滑⽅式，0-贝塞尔平滑  precision :平滑精度，使⽤贝塞尔平滑时⽣效  vamx :设定的最⼤速度，mm/s  amax :设定的最⼤加速度，mm/s2  jmax :设定的最⼤加加速度，mm/s3 | | | |
| 默认参数 | ⽆ | | | | | |
| 返回值 | 错误码 成功-0 失败- errcode | | | | | |

**7.23.** 轨迹复现**(**轨迹前瞻**)**

*在* *python 版本加⼊:* SDK-v2.1.0

|  |  |  |  |
| --- | --- | --- | --- |
| 原型 |  |  |  |
| MoveTrajectoryLA() |
|  | |
| 描述 | 轨迹复现(轨迹前瞻) | | |
| 必选参数 | ⽆ | | |
| 默认参数 | ⽆ | | |
| 返回值 | 错误码 成功-0 失败- errcode | | |



**7.23.1.** 代码示例

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  *条*  15  16  17  18  19  20  21  22 | **from fairino import** Robot  **import time**  *# 与机器⼈控制器建⽴连接，* *连接成功返回—个机器⼈对象*  robot = Robot.RPC( '192.168.58.2')  rtn = 0  rtn = robot.TrajectoryJUpLoad("D://zUP/A.txt")  print("TrajectoryJUpLoad A.txt rtn is ",rtn)  rtn = robot.TrajectoryJUpLoad("D://zUP/B.txt")  print("TrajectoryJUpLoad B.txt rtn is ", rtn)  nameA = "/fruser/traj/A.txt"  nameB = "/fruser/traj/B.txt"  *# rtn = robot.LoadTrajectoryLA(nameA, 2, 0.0, 0, 1.0, 100.0, 200.0, 1000.0) #B样*  *# print("LoadTrajectoryLA rtn is ", rtn)*  robot.LoadTrajectoryLA(nameB, 0, 0, 0, 1, 100, 100, 1000) *#直线连接*  *# robot.LoadTrajectoryLA(nameA, 1, 2, 0, 2, 100, 200, 1000) #直线拟合*  *# error,startPos = robot.GetTrajectoryStartPose(nameA)*  error,startPos = robot.GetTrajectoryStartPose(nameB)  robot.MoveCart(startPos, 1, 0, 100, 100, 100, -1, -1)  rtn = robot.MoveTrajectoryLA()  print("MoveTrajectoryLA rtn is ", rtn) |

