



双臂自碰撞安全检测功能使用手册

Dual-Arm Self-Collision Detection

Version: 1.0.0



目 录

第一章 免责声明	- 1 -
第二章 产品概述	- 1 -
第三章 定义说明	- 1 -
3.1 碰撞体类型	- 1 -
3.2 碰撞对设置	- 2 -
第四章 使用流程	- 3 -
4.1 文件层级	- 3 -
4.2 版本支持	- 3 -
4.3 执行脚本	- 3 -
4.4 可用接口	- 5 -

第一章 免责声明

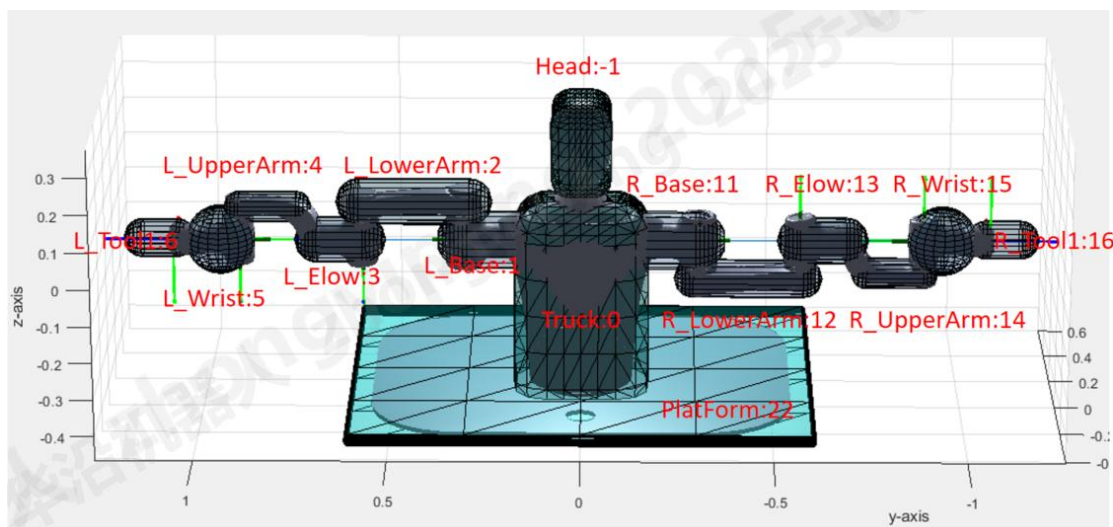
- (1) 零力示教操作：在使用零力示教进行拖拽操作前，用户必须确认负载重量及机器人基座安装角度等参数设置正确。因参数设置不当引发的任何问题，由用户自行承担。
- (2) 系统延迟与点动：当前系统在执行短点动操作时无法瞬时停止，且存在一定的系统延迟。因此，在实际调试过程中，用户有义务主动控制运行速度，确保操作安全。
- (3) 包络体参数设置：包络体保护功能的生效前提是参数设置为有效正值。若用户将相关参数修改为负数，则该保护功能将失效。用户应避免此类设置，并对自身修改参数的行为负责。

第二章 产品概述

本系统是一款面向双机械臂协同作业场景的高可靠性、实时安全防护解决方案。它通过集成的数据通信、实时碰撞检测与毫秒级急停响应，有效预防机械臂之间、机械臂与环境之间的意外碰撞，保障人员安全、设备完整性与生产流程的连续性。

第三章 定义说明

3.1 碰撞体类型



- (1) 棱体模型：用于平台、头部等固定部件（含参考系转换、几何尺寸和偏移参数）。
包含：平台(22)、头部(-1)、躯干(0)。
- (2) 胶囊体模型：用于机械臂连杆（含起点、终点坐标和半径）。
包含：左臂：基座(1)、下臂(2)、肘部(3)、上臂(4)、工具1(6)。
右臂：基座(11)、下臂(12)、肘部(13)、上臂(14)、工具1(16)。

(3) 球体模型：用于腕部关节（含偏移量和半径）。

包含：左腕部(5)、右腕部(15)。

3.2 碰撞对设置

碰撞对类型	碰撞对数量	具体碰撞对(ID1, ID2)
平台(22)相关	10对	(22,1), (22,2), (22,3), (22,4), (22,5), (22,11), (22,12), (22,13), (22,14), (22,15)
头部(-1)相关	10对	(-1,2), (-1,3), (-1,4), (-1,5), (-1,6), (-1,12), (-1,13), (-1,14), (-1,15), (-1,16)
躯干(0)相关	10对	(0,12), (0,13), (0,14), (0,15), (0,16), (6,0), (5,0), (4,0), (3,0), (2,0)
左臂内部碰撞	6对	(5,3), (5,2), (5,1), (4,2), (4,1), (3,1)
右臂内部碰撞	6对	(11,13), (11,14), (11,15), (12,14), (12,15), (13,15)
双臂间碰撞	24对	(5,11), (5,12), (5,13), (5,14), (5,15), (4,11), (4,12), (4,13), (4,14), (4,15), (3,11), (3,12), (3,13), (3,14), (3,15), (2,11), (2,12), (2,13), (2,14), (2,15), (1,12), (1,13), (1,14), (1,15)
工具相关碰撞	17对	(6,2), (6,1), (6,22), (6,11), (6,12), (6,13), (6,14), (6,15), (6,16), (16,12), (16,11), (16,22), (16,1), (16,2), (16,3), (16,4), (16,5)

如：(22,4) 指的是平台 (PlatForm: 22) 与左上臂 (L_UpperArm: 4) 的包络体设置了碰撞检测，如果发生碰撞，会启动急停。

第四章 使用流程

4.1 文件层级

双臂自碰撞安全检测模块/

├─ 双臂机器人产品手册

├─ 双臂自碰撞安全检测功能使用手册

└─ code/

 ├─ demo/

 ├─ collision_detection_server.py (双臂自碰撞服务端)

 ├─ CPS.py (机械臂接口)

 ├─ dual_arm_collision.json (双臂包络体配置文件)

 ├─ dual_arm_collision.so (双臂自碰撞算法库)

 ├─ dualarm_tool_collision.json (夹爪包络体配置文件)

 ├─ readDualArmSelfCollisionModelFromJson (读取配置文件脚本)

 └─ robot_arm_client.py (双臂自碰撞客户端)

 └─ sdk/

 ├─ dual_arm_collision.so

 ├─ 双臂机器人碰撞体定义

 └─ 双臂自碰撞接口文档

4.2 版本支持

(1) 处理器架构：必须使用基于 x86 架构的处理器。其他架构（如 ARM 等）不予兼容。

(2) Python 解释器：必须严格使用 Python 3.8 版本。任何高于或低于此版本的 Python 环境均不被支持。

(3) 环境依赖安装：在部署或运行自碰撞检测算法前，必须严格安装其全部依赖项。

4.3 执行脚本

(1) 连接左右机械臂，使得机械臂和本机处于同一网段。

(2) 打开robot_arm_client.py文件，修改下列代码：

```
def __init__(self, server_host='192.168.1.8', server_port=9092)
```

将server_host修改为本机ip。

(3) `dual_arm_collision.json` 和 `dualarm_tool_collision.json`可自定义双臂以及夹爪的包络体参数。

```
"base": {
  "start": [0, 0, -0.02],
  "end": [0, 0, 0.30],
  "radius": 0.115,
  "type": "capsule",
  "referenceFrame": "base_stand"
},
```

修改-0.02 和 0.3可更改胶囊体的长度，修改 0.115 可更改胶囊体的半径。

```
"r_wrist": {
  "offset": [0, 0, 0.05],
  "radius": 0.11,
  "type": "sphere",
  "referenceFrame": "coor_right4"
},
```

修改 0.11 可更改球体半径。

```
"truck":{
  "ref2local_frame":[0, 0, 0, 0, 0, 0],
  "geometry":[0.12, 0.18, 0.4, 0.09],
  "offset":[0.01, 0, -0.175],
  "type": "lozenge"
},
```

geometry参数指的是：长、宽、高、半径。

(4) 执行指令： `python3 collision_detection_server.py` 启动服务端

```
双机械臂碰撞检测系统初始化成功
Collision detection server started on 0.0.0.0:9092
```

出现“初始化成功”即为启动服务端成功。

(5) 执行指令： `python3 robot_arm_client.py` 启动客户端

```
CPS client initialized
开始连接left arm (boxid: 0, ip: 192.168.1.20, port: 10003)
http://192.168.1.20:20000
left arm连接成功 (boxid: 0, ip: 192.168.1.20)
开始连接right arm (boxid: 1, ip: 192.168.1.30, port: 10003)
http://192.168.1.30:20000
right arm连接成功 (boxid: 1, ip: 192.168.1.30)
Connected to collision detection server at 192.168.1.8:9092
```

192.168.1.20 和 192.168.1.30为左右机械臂ip。

192.168.1.8 为本机ip，可根据自己网段进行修改。

出现“Connected”则为客户端连接服务端成功。

(6) 检测碰撞：

```
Received from ('192.168.1.8', 58204): {'type': 'joint_data', 'joint_positions': [6.283342386812265, -1.4180974705379128, 0.14610151168444532, -0.17465509
824707257, 1.3157513632009652, -0.23904029435314336, 3.0199955646033483, 0.5019043329960093, -1.1678696156794857, -1.2153774779187712, -1.316327321854123
4, 0.12391837689159739], 'joint_velocities': [0.02, -0.0, -0.0, 0.02, -0.0, -0.0, -0.0, -0.0, -0.0, 0.02, -0.0, 0.02], 'units': {'positions': 'radians',
'velocities': 'degrees_per_second'}}
```

```
===== 开始更新机器人坐标 =====

输入关节角度 (12维):
左臂: 6.28334 -1.4181 0.146102 -0.174655 1.31575 -0.23904
右臂: 3.02 0.501904 -1.16787 -1.21538 -1.31633 0.123918

DH参数表 (a, alpha, d, theta):
关节0: 0, 1.5708, 0, 0
关节1: 0, 1.5708, 0.3075, 0
关节2: 0.266, 3.14159, 0, 1.5708
关节3: 0, 1.5708, 0, 1.5708
关节4: 0, -1.5708, 0.324, 0
关节5: 0, 1.5708, 0, 0
关节6: 0, 0, 0.175, 0

实际计算角度:
左臂: 6.28334 0.152699 1.7169 -0.174655 1.31575 -0.23904
右臂: 3.02 2.0727 0.402927 -1.21538 -1.31633 0.123918
左臂基座位置: 0 0 0
右臂基座位置: 0 0 0
左肘位置: 0.262905 0.34796 -4.13104e-05
右肘位置: 0.127027 -0.540694 -0.0155227
左腕位置: 0.586898 0.350098 -9.21931e-05
右腕位置: 0.44706 -0.508677 -0.0546307

最终TCP位置:
左臂TCP: 0.632152 0.18363 0.0293265
右臂TCP: 0.465534 -0.445677 -0.216849
===== 坐标更新完成 =====

碰撞检测结果: False, 碰撞对: [16, 22], 距离: 0.012781452387571335
```

当检测结果为**False**时，说明碰撞对未发生碰撞。

4.4 可用接口

函数接口	功能
check_collision	碰撞检测
check_collision_Pair	检查碰撞对
check_collision_model	查询碰撞模型
dualarm_tagAXISPOS_REF	创建jointPos对象
init_dual_arm	初始化双机械臂
remove_collision_Pair	删除碰撞对
remove_collision_model	删除碰撞模型
set_collision_Pair	设置碰撞对
set_toolcollisionmodel	设置末端工具
update_joints	更新关节参数