
目录

目录.....	1
修订记录.....	2
1. 启动软件.....	3
1.1 进入 ros 工程路径.....	3
1.2 打开 can 通信.....	3
1.3 打开软件界面.....	3
2. 软件功能使用.....	3
2.1 机器人选择界面.....	3
2.2 机器人控制主界面.....	3
2.2.1 关节空间位置控制.....	3
2.2.2 关节空间速度控制.....	4
2.2.3 离线轨迹控制.....	4
2.2.4 笛卡尔空间位置控制.....	5
2.2.5 接收数据.....	6
2.3 机器人工具栏.....	6
2.3.1 夹持器控制.....	6
2.3.2 机器人示教记录.....	7
2.3.3 机器人状态显示.....	7
2.3.4 零点设置.....	7
3. ROS 控制.....	8
3.1 Ros 控制命令.....	8
3.2 Ros 状态反馈.....	9
4. Qt 界面.....	9
4.1 Qt 添加功能界面.....	9
4.2 Qt 机器人信号控制.....	9

修订记录

修订版本	修订时间	作者	备注
V1.0	2020.8.7	谭	初始版本
V1.1	2020.11.13	kiko	新增接收数据

1. 启动软件

1.1 进入 ros 工程路径

终端输入: `cd ~/ros/BirlModuleRobot/`, 若没有采用 `./install.sh` 文件默认路径安装, 此处改为 ros 工程自定义路径。

1.2 打开 can 通信

终端输入: `roslaunch canopen_communication can_prepare.sh`

输入系统密码, 即可通信。若没有连接 can 卡, 将出现如下错误:

Cannot find device "can0"

Cannot find device "can0"

SIOCGIFINDEX: No such device

1.3 打开软件界面

打开新的终端输入: `roslaunch ui ui.launch`

至此, 若无报错, 软件成功打开。

2. 软件功能使用

2.1 机器人选择界面

点击机器人按钮, 进入相应的机器人控制界面。



图 2-1 机器人选择界面

2.2 机器人控制主界面

机器人控制主界面包含关节位置控制、关节速度控制、笛卡尔空间位置控制和离线轨迹四种模式。如图 2-2 所示, 机器人初始状态处于“未使能状态”, 点击启动按钮, 机器人状态将转换为“机器人使能中”, 等待五秒左右, 机器人状态将转换为“机器人使能成功”。另外, 可通过暂停、停止和急停按钮控制机器人运动。

2.2.1 关节空间位置控制

如图 2-2 所示, 机器人默认关节速度为 5 度每秒, 速度调节范围限制在 1~30 度每秒。通过输入各关节对应的关节位置目标, 点击相应的运行按钮, 机器人关节及产生运动。



图 2-2 机器人关节位置控制界面

2.2.2 关节空间速度控制

如图 2-3 所示，机器人默认关节速度为 5 度每秒，速度调节范围限制在-30~30 度每秒。点击相对关节运行按钮，机器人将以设定速度运行。再次点击运行按钮，机器人立刻停止。



图 2-3 机器人关节速度控制界面

2.2.3 离线轨迹控制

如图 2-4 所示，通过复制相应路径点集合（如下述 **example**）到文本框，设置路径最大速度。点击载入文件按钮，界面将弹窗提示路径点格式是否正确。若格式正确，点击运行按钮，机器人将以指定路径运行。



图 2-4 机器人离线轨迹控制界面

路径代码 example:

P=0,0,0,0,0,;

P=0.0,13.654,-0.0,15.605,0.0,;

P=0.0,2.774,-0.0,2.78,0.0,;

P=0.0,-14.26,-0.0,2.78,0.0,;

P=0.0,-14.26,-0.0,-26.912,0.0,;

P=0.0,0.0,-0.0,-0.0,0.0,;

2.2.4 笛卡尔空间位置控制

如图 2-5 所示，笛卡尔空间位置速度命令默认为机器人工作初始位姿。在机器人工作空间范围内，选择合适的笛卡尔位置速度命令；点击获取逆解按钮；窗口将显示关节位置、速度命令，确认安全无误，点击运行按钮。



图 2-5 机器人笛卡尔空间控制界面

2.2.5 接收数据

如图 2-6 所示，点击 Receive Data 按键，管道接收函数启动，只要管道打开且有数据(接收设置为 6 位*(8 位有效数据的 float 数字))，就会在接收笛卡尔位置显示管道接收的数据；点击“位置转换”按钮，便会将该接收到的相对位姿转换到机器人基座标系下的位姿，显示在“到基座笛卡尔位置显示”，并在“转换后是否正确”上显示该转换结果是否可以在机器人工作空间运行。当转换结果为“True”时，显示转换后的关节位置命令和关节速度命令(关节速度命令依赖于在 source code 中设置的笛卡尔速度命令[0.004, 0.002, 0.002, 0.002, 0.004], unit: m/s)。只有在转换结果为“True”的情况下，点击“运行”按钮，机器人才会进行运动。



图 2-6 接收数据界面

2.3 机器人工具栏

2.3.1 夹持器控制

点击工具 → 夹持器。如图 2-7 所示，夹持器电流默认 200mA，右侧文本动态显示夹持器控制电流。点击开启按钮，将使能电流控制打开相应夹持器，再次点击开启按钮，夹持器停止使能电流，即夹持器停止运动。



图 2-7 夹持器控制界面

2.3.2 机器人示教记录

点击 工具→ 示教记录。如图 2-8 所示，机器人运动到待记录位置，点击插入点按钮，左侧文本将显示插入路径点。添加路径点过程中，可通过删除点按钮删除错误采样点。采样路径完毕，通过左下侧文本框输入保存路径文件名，点击保存按钮，文件将保存在 `third_modular_robot/ui/path_point_file/`目录下。



图 2-8 机器人示教记录界面

2.3.3 机器人状态显示

如图 2-9 所示，机器人启动成功后，点击 工具→机器人状态反馈，该界面动态显示笛卡尔位姿、关节位置、关节速度和关节电流状态。



图 2-9 机器人状态显示界面

2.3.4 零点设置

点击 工具→ 零点设置。如图 2-10 所示，点击获取当前位置按钮，关节文本框将更新关节位置。关节运动方向默认为正方向。若机器人关节运动方向与预期相反，可通过正向选择框更改运动方向。确认机器人零点位置后，点击设置按钮设置关节零点。



图 2-10 机器人零点设置界面

3. ROS 控制

3.1 Ros 控制命令

1. 机器人关节控制命令消息类型 `third_modular_robot/ui/msg/robot_command.msg`:

时间戳

Header timeHeader

关节位置命令 I1 T2 T3 T4 I5

float32[] CommandPosData

关节速度命令

float32[] CommandVelData

2. 机器人接收话题: `/low_level/joints_point_command`

采用上述消息类型、话题；控制程序可编写相对应的 ros 话题发布者进行机器人控制。点击工具栏 ROS → 控制命令。如图 2-11 所示，ros 发布者发布机器人控制命令，将显示在文本框中。通过删除按钮，可删除相关错误控制命令。清空按钮可清空接收的所有的控制命令。确认控制命令安全无误后，点击发送控制命令按钮控制机器人运动。



图 2-11 机器人接受 ros 命令界面

3.2 Ros 状态反馈

反馈消息类型，third_modular_robot/ui/msg/robot_feedback.msg:

时间戳

Header timeHeader

反馈关节位置数据 I1 T2 T3 T4 I5

float32[] feedbackPosData

反馈关节速度数据

float32[] feedbackVelData

反馈关节电流数据

float32[] feedbackCurrData

ros 状态反馈系统默认关闭状态，通过工具栏 ROS → 勾选状态反馈即可打开。

4. Qt 界面

4.1 Qt 添加功能界面

1. 使用 qtcreator 新建界面文件，添加或删除控件。

2. 将 ui 文件转为 python 文件。

命令: `sudo pyuic5 -o 输出 python 文件 ui 源文件。`

示例: `sudo pyuic5 -o modular_robot_control.py modular_robot_control.ui`

生成的 *.py 放在 third_modular_robot/ui/script, *.ui 放在 third_modular_robot/ui/ui_file/.

4.2 Qt 机器人信号控制

信号位置: third_modular_robot/ui/script/modular_robot_control_func.py

1. 机器人关节位置模式控制信号: `sin_joint_position = pyqtSignal(list)`

信号内容格式:

Python 列表: `[temp_pos_command, temp_velocity_command]`

`temp_pos_command`: 关节控制位置列表 (I1, T2, T3, T4, I5), 单位角度。

`temp_velocity_command`: 关节控制速度列表 (I1, T2, T3, T4, I5), 单位度每秒。

2 机器人关节速度模式控制信号: `sin_joint_velocity = pyqtSignal(list)`

信号内容格式:

Python 列表: 关节控制速度列表 (I1, T2, T3, T4, I5), 单位度每秒。