chemicalrobot_arm

1.0.0

制作者 Doxygen 1.8.17

1 索引

1.1 列表

里列出了所有、构、合以及接口定等，并附要明:

2 明

2.1 dashboardsrv_client 参考

Public 成函数

- **dashboardsrv_client** (ros::NodeHandle nh)
- void load_program (const string filename)
  入并后程序
- void play ()
  运行加的程序
- void stop ()
  停止程序
- void robot_init ()
  打*UR5e*源并放制
- void DO_init ()
  将数字出引脚置*0*
- void setDO (int8_t pin, bool state)
  置数字出引脚的平
- void unlockPS ()
  解除保性停止
- void RG2grip ()
  已弃用
- void RG2release ()
  已弃用
- bool RG2_is_busy ()
  已弃用
- bool RG2_is_grip ()
  已弃用
- int close ()
  催化舵机
- int open ()
  打催化舵机
- int grasp ()
  气爪合
- int loose ()
  气爪打
- int reset ()
  气爪位

2.1.1 成函数明

2.1.1.1 close() `int dashboardsrv_client::close ( )`

催化舵机

返回

int

2.1.1.2 DO_init() void dashboardsrv_client::DO_init ( )

将数字出引脚置0

2.1.1.3 grasp() int dashboardsrv_client::grasp ( )

气爪合

返回

int

2.1.1.4 load_program() void dashboardsrv_client::load_program (const string *filename*)

入并启程序

参数

filename	程序名称

2.1.1.5 loose() int dashboardsrv_client::loose ( )

气爪打

返回

int

2.1.1.6 open() int dashboardsrv_client::open ( )

打催化舵机

返回

int

2.1.1.7 play() void dashboardsrv_client::play ( )

运行加的程序

2.1.1.8 reset() int dashboardsrv_client::reset ( )

气爪位

返回

int

2.1.1.9 RG2_is_busy() bool dashboardsrv_client::RG2_is_busy ( )

已弃用

2.1.1.10 RG2_is_grip() bool dashboardsrv_client::RG2_is_grip ( )

已弃用

2.1.1.11 RG2grip() void dashboardsrv_client::RG2grip ( )

已弃用

2.1.1.12 RG2release() void dashboardsrv_client::RG2release ( )

已弃用

2.1.1.13 robot_init() void dashboardsrv_client::robot_init ( )

打UR5e源并放制

2.1.1.14 setDO() void dashboardsrv_client::setDO (

```
int8_t pin,
bool state )
```

置数字出引脚的平

参数

pin	引脚的位置
state	置的平

2.1.1.15 stop() void dashboardsrv_client::stop ( )

停止程序

2.1.1.16 unlockPS() void dashboardsrv_client::unlockPS ( )

解除保性停止

的文档由以下文件生成:

- dashboard.h
- dashboard.cpp

2.2 F710 参考

技F710手柄(用于将/joy信息映射到按)

Public 属性

- int x
  按
- int **a**
- int **b**
- int **y**
- int **LB**
- int **RB**
- int **LT**
- int **RT**
- int **BACK**
- int **unknown1**
- int **unknown2**
- int **unknown3**
- float button_l
  杆
- float **button_up**
- float **rstick_up**
- float **rstick_l**
- float **lstick_up**
- float **lstick_l**

2.2.1 描述

技F710手柄(用于将/joy信息映射到按)

2.2.2 成量明

2.2.2.1 **button_l** `float F710::button_l`

杆

2.2.2.2 **x** `int F710::x`

按

的文档由以下文件生成:

- joystick_control.cpp

2.3 **hole_cls** 参考

离心机孔位信息

Public 成函数

- **hole_cls** (int num)

Public 属性

- Isometry3d **_mTh**
- vector< int > **_fix_seq1**
- vector< int > **_fix_seq2**
- double **_fix_angle**

2.3.1 描述

离心机孔位信息

的文档由以下文件生成:

- record.cpp

2.4 myException 参考

异常

```
#include <motion_planning.h>
```

Public 成函数

- **myException** (vector< string > msgs={"YES", ""}, vector< double > relocation_err={}, vector< int > exception_bottle={}, int rack_absnum=0)
- std_msgs::String **what** ()
- string **getException** ()

Public 属性

- Json::Value **root1**
- Json::Value **root2**
- std_msgs::String **_pubmsgs**

2.4.1 描述

异常

的文档由以下文件生成:

- motion_planning.h
- motion_planning_util.cpp

2.5 ur5e_action 参考

机器人化学家机械臂部分主要模

```
#include <motion_planning.h>
```

Public 成函数

- **ur5e_action** (ros::NodeHandle nh, shared_ptr< dashboardsrv_client > dbptr, moveit::planning_interface::↩ MoveGroupInterfacePtr mgptr)
- void LoadStationInfo (string station_name)

  入特定站点的构化信息（角和位姿）(注意：函数只在机械臂位后才会用，所以需要在站点操作前送位指令)
- void LoadRobotJointInfo ()

  入机器人平台上的角
- void ForceFeedback (string name, double speed, double thre)

  力反控制
- void Init (const string filename)

  初始化*UR5e*和爪,包括后源,放制器,加程控制程序(函数要求*UR5e*于程控制模式)
- bool GotoDefaultPose (string name)

将机械臂移到默位姿

- void SaveStationInfo (std::string station, bool isJoint)

  站点信息主函数,包含*Save*模式和*Go*模式*(TODO)*

- void SwitchMode (int mode)

  用于切模式*(TODO)*

- void SaveMode ()

  站点信息中的*Save*模式,此模式下入名称会用*SaveStationJoint()*或*SaveStationPose()*来保存信息至*json*文件.*(TODO)*

- void GoMode ()

  站点信息中的*Go*模式,此模式下入角或位姿名称就可以移到位置.*(TODO)*

- void SaveStationPose (string file_path, string name, Vector3d X, Quaterniond Q)

  当前位姿到*json*文件中*(TODO)*

- void SaveStationJoint (string file_path, string name, vector< double > joints)

  当前角到*json*文件中*(TODO)*

- void CommandExecute (std::string station, std::string operation, int rack, int bottle, int rack_oper, int light_num, int paper, int dryer_num, int libs_num)

  解析并行任度模送的指令*(TODO)*

- void MoveLine_pilz (Movetowards towards, double dis, double speed_factor)

  机械臂末端在工作空中沿直迹移，会在起点和目之按照一条直运

- void MoveCircle_pilz (CirType type, std::string name, std::string goal)

  机械臂末端在工作空中沿弧迹运，需要指定心或中点*(TODO)*

- void SequenceMove_pilz (vector< string > &V_name, vector< double > &V_radius, vector< double > &V↩ _speed, vector< double > &V_acc, vector< string > &V_planner, bool validate, bool record)

  根据定的操作行移

- void SequenceMove_pilz (vector< string > &V_name, vector< double > &V_radius, vector< double > &V_↩ speed, vector< double > &V_acc, vector< string > &V_planner, bool validate, bool record, std::string speed-profile)

  根据定的操作行移*(TODO)*

- int32_t ValidateOneReq (vector< string > &V_name, vector< double > &V_radius, vector< double > &V_↩ speed, vector< double > &V_acc, vector< string > &V_planner)

  定的个操作是否可行*(*操作在必要会插入路径点*)*

- void SequenceValidate_pilz (vector< vector< string > * > V_name, vector< vector< double > * > V_radius, vector< vector< double > * > V_speed, vector< vector< double > * > V_acc, vector< vector< string > * > V_planner)

  定的多个操作是否可行

- moveit_msgs::MotionSequenceRequest ConstructSeqReq (vector< string > &V_name, vector< double > &V_radius, vector< double > &V_speed, vector< double > &V_acc, vector< string > &V_planner, bool validate)

  根据定的操作，构造迹划的*request*

- void ResetArm ()

  异常情况下位机械臂。函数会倒序行存的路径点和爪操作，直到末端没有持物体且到达安全位置。若在恢程中出异常，会停止运并死机械臂点。此需要人工介入。

- void Gripper_op (uint16_t position, bool check=false, bool block=false, uint16_t force=20, uint16_t speed=50)

  控制爪合和,并是否持有物体

- void JsonPub (vector< string > msgs={"YES", ""}, vector< double > relocation_err={}, vector< int > exception_bottle={}, int rack_absnum=0)

  操作状反,向上位机布*/obsOperation_out*

- void setAvgCartesianSpeed (moveit::planning_interface::MoveGroupInterface::Plan &plan, const string end↩ _effector, const double speed)

  定笛卡迹的末端速度,已弃用

- Isometry3d CalculatePoses (string station_name)

  根据当前站点的位姿算相基座位姿*(*用于重定位之后*)*

- bool InsertPoses (vector< string > &V_name)

根据定的路径点序列插入路径点

- void obs_Callback (const std_msgs::StringConstPtr &obsOperation_in)

  */obsOperation_in*的回函数,接收上位机指令行操作
- void safety_Callback (const ur_dashboard_msgs::SafetyModeConstPtr &safetymode)

  */ur_hardware_interface/safety_mode*的回函数,解除保性停止
- void exec_Callback (const moveit_msgs::ExecuteTrajectoryActionResultConstPtr &execresult)

  */execute_trajectory/result*的回函数,取迹行果
- void RecordSequence (size_t &i, vector< string > V_name, vector< double > V_radius, vector< double > V_speed, vector< double > V_acc, vector< string > V_planner)

  上一次迹划的信息
- void RecordCurrSequence (double speed, double acc, string planner)

  当前位置到下一位置的迹划信息
- bool ClearSequence ()

  根据持状和当前末端位置清空已的路径点
- void GetNextpose (string name, moveit_msgs::Constraints &pose_goal, geometry_msgs::PoseStamped &next_pose)

  根据指定名称,算下一个目位置和束
- double GetBlendradius (geometry_msgs::Pose curr_pose, geometry_msgs::Pose next_pose, geometry_msgs::Pose n_next_pose)

  根据当前位置,目位置和下一位置算融合半径
- vector< double > GetJoints (string name)

  从*_Init_joints*和*_Robot_joints*中取角

Public 属性

- moveit::planning_interface::MoveGroupInterfacePtr _mgptr

  moveit movegroup 指
- shared_ptr< dashboardsrv_client > _dbptr

  *UR5e*的*dashboard*客端,用于程后*UR5e*,加程控制程序
- string _file_root

  存放站点位姿信息文件的路径
- string _curr_station

  机器人当前站点
- geometry_msgs::Pose _CTF_pose

  离心机位姿

2.5.1 描述

机器人化学家机械臂部分主要模

2.5.2 成函数明

2.5.2.1 CalculatePoses() `Isometry3d ur5e_action::CalculatePoses ( string station_name )`

根据当前站点的位姿算相基座位姿(用于重定位之后)

参数

station_name	当前站点名称

返回

位姿

2.5.2.2 CommandExecute() void ur5e_action::CommandExecute (

```
std::string station,
std::string operation,
int rack,
int bottle,
int rack_oper,
int light_num,
int paper,
int dryer_num,
int libs_num )
```

解析并行任度模送的指令(TODO)

参数

station	工作站名称
operation	操作名称
rack	机器人平台上的管架号
bottle	管架上的管号
rack_oper	起始品架,回收品架的管架号
light_num	光催化位置号
paper	催化碳号
dryer_num	烘干机位置号
libs_num	libs品号

2.5.2.3 ConstructSeqReq() moveit_msgs::MotionSequenceRequest ur5e_action::ConstructSeqReq (

```
vector< string > & V_name,
vector< double > & V_radius,
vector< double > & V_speed,
vector< double > & V_acc,
vector< string > & V_planner,
bool validate )
```

根据定的操作，构造迹划的request

参数

V_name	需要的路点名称
V_radius	个路点的融合半径系数(0,1)(最后一个路径点必0)
V_speed	个路点的速度系数(0,1]
V_acc	个路点的加速度系数(0,1]
V_planner	个路点的划器(PTP,LIN)
validate	是否操作是否可行

返回

moveit_msgs::MotionSequenceRequest 构造得到的迹划request

2.5.2.4 ForceFeedback()

```
void ur5e_action::ForceFeedback (
        string name,
        double speed,
        double thre )
```

力反控制

参数

name	目位姿
speed	移速度系数，建0.01，以便有足反
thre	力矩，超此会停止移

2.5.2.5 GetBlendradius()

```
double ur5e_action::GetBlendradius (
        geometry_msgs::Pose curr_pose,
        geometry_msgs::Pose next_pose,
        geometry_msgs::Pose n_next_pose )
```

根据当前位置,目位置和下一位置算融合半径

参数

curr_pose	当前位置
next_pose	目位置
n_next_pose	目位置的下一位置

2.5.2.6 GetJoints()

```
vector< double > ur5e_action::GetJoints (
```

```
            string name )
```

从_Init_joints和_Robot_joints中取角

参数

角的名字	

2.5.2.7 GetNextpose()

```
void ur5e_action::GetNextpose (
            string name,
            moveit_msgs::Constraints & pose_goal,
            geometry_msgs::PoseStamped & next_pose )
```

根据指定名称,算下一个目位置和束

参数

name	指定路点名,"J"角,"P"位置
pose_goal	需要构造的目束
next_pose	需要构造的目位置

2.5.2.8 GoMode()

```
void ur5e_action::GoMode ( )
```

站点信息中的Go模式,此模式下入角或位姿名称就可以移到位置.(TODO)

2.5.2.9 GotoDefaultPose()

```
bool ur5e_action::GotoDefaultPose (
            string name )
```

将机械臂移到默位姿

参数

name	指定需要移的角

2.5.2.10 Gripper_op()

```
void ur5e_action::Gripper_op (
            uint16_t position,
            bool check = false,
            bool block = false,
```

```
        uint16_t force = 20,
        uint16_t speed = 50 )
```

控制爪合和,并是否持有物体

参数

position	爪合位置(0:完全打,1000:完全合)
check	是否持到物体
block	是否等待爪完全到位
force	定抓取力(0,100]
speed	定合速度(0,100]

2.5.2.11 Init() `void ur5e_action::Init (`

```
        const string filename )
```

初始化UR5e和爪,包括启源,放制器,加程控制程序(函数要求UR5e于程控制模式)

参数

filename	用于指定加哪个程序,目前只需要加”remote_control.urp\n”

2.5.2.12 InsertPoses() `bool ur5e_action::InsertPoses (`

```
        vector< string > & V_name )
```

根据定的路径点序列插入路径点

参数

V_name	待插入的路径点序列

返回

是否插入了路径点

2.5.2.13 JsonPub() `void ur5e_action::JsonPub (`

```
        vector< string > msgs = {"YES", ""},
        vector< double > relocation_err = {},
        vector< int > exception_bottle = {},
        int rack_absnum = 0 )
```

操作状反,向上位机布/obsOperation_out

参数

msgs	包含IsDone和Exception
relocation_err	底定位差{位置x,y,z,姿x,y,z,w}
exception_bottle	管架中没有管的位置,若Exception"bottle_not_all",参数非空(未使用)
rack_absnum	后区完成后,送管架号(未使用)

2.5.2.14 LoadStationInfo()

```
void ur5e_action::LoadStationInfo (
    string station_name )
```

入特定站点的构化信息（角和位姿）(注意：函数只在机械臂位后才会用，所以需要在站点操作前送位指令)

参数

station_name	需要入的站点名称

2.5.2.15 MoveCircle_pilz()

```
void ur5e_action::MoveCircle_pilz (
    CirType type,
    std::string name,
    std::string goal )
```

机械臂末端在工作空中沿弧迹运，需要指定心或中点(TODO)

参数

type	指定弧的型。CENTER：指定心，会在起点和目之按照短的弧运；INTERIM：指定中点，弧迹会指定的中点
name	弧的心名称或中点名称
goal	目点名称

2.5.2.16 MoveLine_pilz()

```
void ur5e_action::MoveLine_pilz (
    Movetowards towards,
    double dis,
    double speed_factor )
```

机械臂末端在工作空中沿直迹移，会在起点和目之按照一条直运

参数

towards	移方向
dis	移距离(m)
speed_factor	移速度放系数(0,1]

2.5.2.17 RecordCurrSequence()

```
void ur5e_action::RecordCurrSequence (
        double speed,
        double acc,
        string planner )
```

当前位置到下一位置的迹划信息

参数

speed	当前行迹的速度
acc	当前行迹的加速度
planner	当前行迹的划器

2.5.2.18 RecordSequence()

```
void ur5e_action::RecordSequence (
        size_t & i,
        vector< string > V_name,
        vector< double > V_radius,
        vector< double > V_speed,
        vector< double > V_acc,
        vector< string > V_planner )
```

上一次迹划的信息

参数

i	下一个目路径点的索引
V_name	当前行迹的路点名称
V_radius	当前行迹的混合半径
V_speed	当前行迹的速度
V_acc	当前行迹的加速度
V_planner	当前行迹的划器

2.5.2.19 ResetArm()

```
void ur5e_action::ResetArm ( )
```

异常情况下位机械臂。函数会倒序行存的路径点和爪操作，直到末端没有持物体且到达安全位置。若在恢程中出异常，会停止运并死机械臂点。此需要人工介入。

2.5.2.20 SaveMode()

```
void ur5e_action::SaveMode ( )
```

站点信息中的Save模式,此模式下入名称会用SaveStationJoint()或SaveStationPose()来保存信息至json文件.(TODO)

2.5.2.21 SaveStationInfo()

```
void ur5e_action::SaveStationInfo (
        std::string station,
        bool isJoint )
```

站点信息主函数,包含Save模式和Go模式(TODO)

参数

station	站点名称
isJoint	角是位姿

2.5.2.22 SaveStationJoint()

```
void ur5e_action::SaveStationJoint (
        string file_path,
        string name,
        vector< double > joints )
```

当前角到json文件中(TODO)

参数

file_path	需要写入的json文件路径
name	角名称
joints	角数

2.5.2.23 SaveStationPose()

```
void ur5e_action::SaveStationPose (
        string file_path,
        string name,
        Vector3d X,
        Quaterniond Q )
```

当前位姿到json文件中(TODO)

参数

file_path	需要写入的json文件路径
name	位姿名称
X	位姿平移
Q	位姿旋四元数

2.5.2.24 SequenceMove_pilz() [1/2]

```
void ur5e_action::SequenceMove_pilz (
        vector< string > & V_name,
```

```
vector< double > & V_radius,
vector< double > & V_speed,
vector< double > & V_acc,
vector< string > & V_planner,
bool validate,
bool record )
```

根据定的操作行移

参数

V_name	需要的路点名称
V_radius	指定个路点的融合半径系数(0,1)(最后一个路径点必0)。如下所示,真融合半径=融合半径系数∗min(||Pm-1Pm||,||PmPm+1||)
V_speed	指定个路点的速度放系数(0,1]。运速度=速度放系数∗最大运速度
V_acc	指定个路点的加速度放系数(0,1]。运加速度=加速度放系数∗最大运加速度
V_planner	指定个路点的划器(PTP,LIN)。PTP点到点划，LIN直划
validate	是否只是迹可行（参数已被抛弃）
record	是否末端到达了哪个路径点

2.5.2.25 SequenceMove_pilz() [2/2] void ur5e_action::SequenceMove_pilz (

```
vector< string > & V_name,
vector< double > & V_radius,
vector< double > & V_speed,
vector< double > & V_acc,
vector< string > & V_planner,
bool validate,
bool record,
std::string speedprofile )
```

根据定的操作行移(TODO)

参数

V_name	需要的路点名称
V_radius	指定个路点的融合半径系数(0,1)(最后一个路径点必0)。如下所示,真融合半径=融合半径系数∗min(||Pm-1Pm||,||PmPm+1||)
V_speed	指定个路点的速度放系数(0,1]。运速度=速度放系数∗最大运速度
V_acc	指定个路点的加速度放系数(0,1]。运加速度=加速度放系数∗最大运加速度
V_planner	指定个路点的划器(PTP,LIN,CRIC)。PTP点到点迹划，LIN直迹划，CIRC弧迹划
validate	是否只是迹可行（参数已被抛弃）
record	是否末端到达了哪个路径点
speedprofile	速度廓曲(trap,doubleS)。trap梯形速度划，doubleS双S形速度划

2.5.2.26 SequenceValidate_pilz() void ur5e_action::SequenceValidate_pilz (

```
vector< vector< string > * > V_name,
vector< vector< double > * > V_radius,
vector< vector< double > * > V_speed,
vector< vector< double > * > V_acc,
vector< vector< string > * > V_planner )
```

定的多个操作是否可行

参数

V_name	多个需要的路点名称
V_radius	多个路点的融合半径系数(0,1)(最后一个路径点必0)
V_speed	多个路点的速度系数(0,1]
V_acc	多个路点的加速度系数(0,1]
V_planner	多个路点的划器(PTP,LIN)

2.5.2.27 setAvgCartesianSpeed()

```
void ur5e_action::setAvgCartesianSpeed (
    moveit::planning_interface::MoveGroupInterface::Plan & plan,
    const string end_effector,
    const double speed )
```

定笛卡迹的末端速度,已弃用

参数

plan	需要定速度的plan
end_effector	指定末端link
speed	定的末端速度(m/s)

2.5.2.28 SwitchMode()

```
void ur5e_action::SwitchMode (
    int mode )
```

用于切模式(TODO)

参数

mode	0Save模式(默),1Go模式.

2.5.2.29 ValidateOneReq()

```
int32_t ur5e_action::ValidateOneReq (
    vector< string > & V_name,
    vector< double > & V_radius,
```

```
vector< double > & V_speed,
vector< double > & V_acc,
vector< string > & V_planner )
```

定的个操作是否可行(操作在必要会插入路径点)

参数

V_name	需要的路点名称
V_radius	个路点的融合半径系数(0,1)(最后一个路径点必0)
V_speed	个路点的速度系数(0,1]
V_acc	个路点的加速度系数(0,1]
V_planner	个路点的划器(PTP,LIN)
validate	是否操作是否可行

2.5.3 成量明

2.5.3.1 _CTF_pose geometry_msgs::Pose ur5e_action::_CTF_pose

离心机位姿

2.5.3.2 _curr_station string ur5e_action::_curr_station

机器人当前站点

2.5.3.3 _dbptr shared_ptr<dashboardsrv_client> ur5e_action::_dbptr

UR5e的dashboard客端,用于程启UR5e,加程控制程序

2.5.3.4 _file_root string ur5e_action::_file_root

存放站点位姿信息文件的路径

2.5.3.5 _mgptr `moveit::planning_interface::MoveGroupInterfacePtr ur5e_action::_mgptr`

moveit movegroup 指

的文档由以下文件生成:

- motion_planning.h
- motion_planning.cpp
- motion_planning_util.cpp

2.6 ur5e_js 参考

手柄控制UR5e机械臂

Public 成函数

- ur5e_js (ros::NodeHandle nh)
  切*ros controllertwist_controller*
- ~ur5e_js ()
  析构,并切*ros controllerscaled_joint_trajectory_controller*
- void run ()
  遥控主程序

2.6.1 描述

手柄控制UR5e机械臂

平移:x,bx方向平移;y,ay方向平移;LB,RBz方向平移;旋:左右方向x旋;上下方向y旋;LT,RTz旋;爪控制:左遥控左右控制合位置

2.6.2 构造及析构函数明

2.6.2.1 ur5e_js() `ur5e_js::ur5e_js ( ros::NodeHandle nh )`

切ros controllertwist_controller

参数

nh	

2.6.2.2 ~ur5e_js() `ur5e_js::~ur5e_js ( )`

析构,并切ros controllerscaled_joint_trajectory_controller

2.6.3 成函数明

2.6.3.1 run() void ur5e_js::run ( )

遥控主程序

的文档由以下文件生成:

- joystick_control.cpp

Index