**了解API接口**

参考文件：Dobot Magician API

**CP功能：连续运动轨迹**

**设置CP运动的速度和加速度**

**原型：**

int SetCPParams(CPParams \*cpParams, bool isQueued, uint64\_t \*queuedCmdIndex)

**参数**

CPParams 定义：

typedef struct tagCPParams {

float planAcc; //规划加速度最大值

float junctionVel; //拐角速度最大值

union {

float acc; //实际加速度最大值，非实时模式时有效

float period; //插补周期，实时模式时有效

};

uint8\_t realTimeTrack; //0：非实时模式：所有指令下发后再运行

1：实时模式：边下发指令边运行

}CPParams;

cpParams：CPParams 指针

isQueued：是否将该指令指定为队列命令

queuedCmdIndex：若选择将指令加入队列，则表示指令在队列的索引号。

否则，该参数无意义

**返回** DobotCommunicate\_NoError：指令正常返回

DobotCommunicate\_BufferFull：指令队列已满

DobotCommunicate\_Timeout：指令无返回，导致超时

**执行 CP 指令**

**原型** int SetCPCmd(CPCmd \*cpCmd, bool isQueued, uint64\_t \*queuedCmdIndex)

**功能** 执行 CP 指令

**参数**

CPCmd 定义：

typedef struct tagCPCmd {

uint8\_t cpMode; //CP 模式。0：相对模式，表示相对距离，即笛

卡尔坐标增量。1：绝对模式，表示绝对距离，

即笛卡尔坐标系下目标点坐标

float x; //x,y,z //可以设置为坐标增量，也可设置为目的坐

标点

float y;

float z;

union {

float velocity; //保留

float power; //保留

};

}CPCmd;

cpCmd：CPCmd 指针

isQueued：是否将该指令指定为队列命令

queuedCmdIndex：若选择将指令加入队列，则表示指令在队列的索引号。

否则，该参数无意义

**返回** DobotCommunicate\_NoError：指令正常返回

DobotCommunicate\_BufferFull：指令队列已满

DobotCommunicate\_Timeout：指令无返回，导致超时

**设置PTP模式下关节坐标轴的速度和加速度**

**原型** int SetPTPJointParams(PTPJointParams \*ptpJointParams, bool isQueued,

uint64\_t \*queuedCmdIndex)

**功能** 设置 PTP 运动时各关节坐标轴的速度（°/s）和加速度（°/s

2）

**参数**

PTPJointParams 定义：

typedef struct tagPTPJointParams{

float velocity[4]; //PTP 模式下 4 轴关节速度

float acceleration[4]; //PTP 模式下 4 轴关节加速度

}PTPJointParams;

ptpJointParams：PTPJointParams 指针

isQueued：是否将该指令指定为队列命令

queuedCmdIndex：若选择将指令加入队列，则表示指令在队列的索引号。

否则，该参数无意义

**返回** DobotCommunicate\_NoError：指令正常返回

DobotCommunicate\_BufferFull：指令队列已满

DobotCommunicate\_Timeout：指令无返回，导致超时

**设置PTP模式下各笛卡尔坐标轴的速度和加速度**

**原型** int SetPTPCoordinateParams(PTPCoordinateParams \*ptpCoordinateParams, bool

isQueued, uint64\_t \*queuedCmdIndex)

**功能** 设置 PTP 运动时各笛卡尔坐标轴的速度（mm/s）和加速度（mm/s

2）

**参数**

PTPCoordinateParams 定义：

Dobot Magician API 接口说明 12PTP 功能

文档版本 V1.2.3（2019-07-19） 接口说明 版权所有 © 越疆科技有限公司

35

获取 PTP 模式下各笛卡尔坐标轴的速度和加速度

表 12.4 获取 PTP 运动时各笛卡尔关节坐标轴的速度 s 和加速度

设置 JUMP 模式下抬升高度和最大抬升高度

表 12.5 设置 JUMP 模式下抬升高度和最大抬升高度

typedef struct tagPTPCoordinateParams {

float xyzVelocity; //PTP 模式下 X,Y,Z 3 轴坐标轴速度

float rVelocity; //PTP 模式下末端 R 轴速度

float xyzAcceleration; //PTP 模式下 X,Y,Z 3 轴坐标轴加速度

float rAccleration; //PTP 模式下末端 R 轴加速度

} PTPCoordinateParams;

ptpCoordinateParams：PTPCoordinateParams 指针

isQueued:是否将该指令指定为队列命令

queuedCmdIndex：若选择将指令加入队列，则表示指令在队列的索引号。

否则，该参数无意义

**返回** DobotCommunicate\_NoError：指令正常返回

DobotCommunicate\_BufferFull：指令队列已满

DobotCommunicate\_Timeout：指令无返回，导致超时

**设置JUMP模式下抬升高度和最大抬升高度**

**原型** int SetPTPJumpParams(PTPJumpParams \*ptpJumpParams, bool isQueued,

uint64\_t \*queuedCmdIndex)

**参数**

PTPJumpParams 定义：

typedef struct tagPTPJumpParams {

float jumpHeight; //抬升高度

float zLimit; //最大抬升高度

}PTPJumpParams;

ptpJumpParams：PTPJumpParams 指针

isQueued：是否将该指令指定为队列命令

queuedCmdIndex：若选择将指令加入队列，则表示指令在队列的索引号。

否则，该参数无意义

**返回** DobotCommunicate\_NoError：指令正常返回

DobotCommunicate\_BufferFull：指令队列已满

DobotCommunicate\_Timeout：指令无返回，导致超时

**设置PTP运动的速度百分比和加速度百分比**

**原型** int SetPTPCommonParams(PTPCommonParams \*ptpCommonParams, bool

isQueued, uint64\_t \*queuedCmdIndex)

**功能** 设置 PTP 运动的速度百分比和加速度百分比

**参数**

PTPCommonParams 定义：

typedef struct tagPTPCommonParams {

float velocityRatio; //PTP 模式速度百分比，关节坐标轴和笛卡尔

坐标轴共用

float accelerationRatio; //PTP 模式加速度百分比，关节坐标轴和笛卡尔

坐标轴共用

}PTPCommonParams;

ptpCommonParams：PTPCommonParams 指针

isQueued：是否将该指令指定为队列命令

queuedCmdIndex：若选择将指令加入队列，则表示指令在队列的索引号。否

则，该参数无意义

**返回** DobotCommunicate\_NoError：指令正常返回

DobotCommunicate\_BufferFull：指令队列已满

DobotCommunicate\_Timeout：指令无返回，导致超时**执行PTP指令**

**原型** int SetPTPCmd(PTPCmd \*ptpCmd, bool isQueued, uint64\_t \*queuedCmdIndex)

**功能** 执行 PTP 指令。设置 PTP 相关参数后，调用此函数可使机械臂运动至设置的

目标点。

**参数**

PTPCmd 定义：

typedef struct tagPTPCmd {

uint8\_t ptpMode; //PTP 模式，取值范围：0~9

float x; //（x,y,z,r）为坐标参数，可为笛卡尔坐标、

节坐标、笛卡尔坐标增量或关节坐标增量

float y;

float z;

float r;

}PTPCmd;

//其中，ptpMode 取值如下：

enum {

JUMP\_XYZ, //JUMP 模式，（x,y,z,r）为笛卡尔坐标系下的目标

点坐标

MOVJ\_XYZ, //MOVJ 模式，（x,y,z,r）为笛卡尔坐标系下的目标

点坐标

MOVL\_XYZ, //MOVL 模式，（x,y,z,r）为笛卡尔坐标系下的目标

点坐标

JUMP\_ANGLE, //JUMP 模式，（x,y,z,r）为关节坐标系下的目标点

坐标

MOVJ\_ANGLE, //MOVJ 模式，（x,y,z,r）为关节坐标系下的目标点

坐标

MOVL\_ANGLE, //MOVL 模式，（x,y,z,r）为关节坐标系下的目标

点坐标

MOVJ\_INC, //MOVJ 模式，（x,y,z,r）为关节坐标系下的坐标

增量

MOVL\_INC, //MOVL 模式，（x,y,z,r）为笛卡尔坐标系下的坐

标增量

MOVJ\_XYZ\_INC, //MOVJ 模式，（x,y,z,r）为笛卡尔坐标系下的坐

标增量

JUMP\_MOVL\_XYZ, //JUMP 模式，平移时运动模式为 MOVL。

（x,y,z,r）为笛卡尔坐标系下的坐标增量

};

ptpCmd：PTPCmd 指针

isQueued：是否将该指令指定为队列命令

queuedCmdIndex：若选择将指令加入队列，则表示指令在队列的索引号。

否则，该参数无意义

**返回** DobotCommunicate\_NoError：指令正常返回

DobotCommunicate\_BufferFull：指令队列已满

DobotCommunicate\_Timeout：指令无返回，导致超时

**PTP功能**：实现点到点的运动，其中大致共有三种运动模式

1 MOVJ模式：关节运动

2 MOVL模式：直线运动

3 JUMP模式：门型轨迹

区分MOVJS和MOVL

前者为关节运动，即在运动的过程中关节角发生变化，特征为各个关节轴运行时间一致，而且同时到达终点

后者为普通且简单的点到点的直线运动

**机械臂再现运动时各轴实际运动速度与设置的速度关系如下所示:**

各轴再现速度=各轴设置的再现速度\*设置的速度百分⽐

各轴再现加速度=各轴设置的再现加速度\*设置的加速度百分⽐