



(12)发明专利申请

(10)申请公布号 CN 106945044 A

(43)申请公布日 2017.07.14

(21)申请号 201710256933.0

(22)申请日 2017.04.19

(71)申请人 广州视源电子科技股份有限公司

地址 510530 广东省广州市广州黄埔区云
埔四路6号

(72)发明人 阳方平

(74)专利代理机构 广州华进联合专利商标代理
有限公司 44224

代理人 余永文

(51)Int.Cl.

B25J 9/16(2006.01)

B25J 13/08(2006.01)

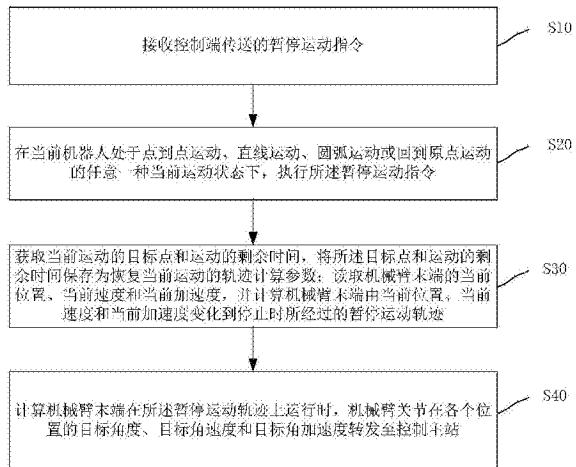
权利要求书5页 说明书16页 附图5页

(54)发明名称

机器人暂停运动控制方法和系统

(57)摘要

本发明涉及一种机器人暂停运动控制方法和系统，所述方法包括：接收控制端传送的暂停运动指令；在当前机器人处于点到点运动、直线运动、圆弧运动或回到原点运动的任意一种当前运动状态下；获取当前运动的目标点和运动的剩余时间，将目标点和运动的剩余时间保存为恢复当前运动的轨迹计算参数；读取机械臂末端的当前位置、当前速度和当前加速度，并计算机械臂末端由当前位置、当前速度和当前加速度变化到停止时所经过的暂停运动轨迹；计算机械臂末端在暂停运动轨迹上运行时，机械臂关节在各个位置的目标角度、目标角速度和目标角加速度转发至控制主站；本发明可以构成一个完整的机器人暂停运动控制系统，降低机器人控制系统开发成本，提高控制效果。



1. 一种机器人暂停运动控制方法,其特征在于,包括:

接收控制端传送的暂停运动指令;

在当前机器人处于点到点运动、直线运动、圆弧运动或回到原点运动的任意一种当前运动状态下,执行所述暂停运动指令;

获取当前运动的目标点和运动的剩余时间,将所述目标点和运动的剩余时间保存为恢复当前运动的轨迹计算参数;读取机械臂末端的当前位置、当前速度和当前加速度,并计算机械臂末端由当前位置、当前速度和当前加速度变化到停止时所经过的暂停运动轨迹;

计算机械臂末端在所述暂停运动轨迹上运行时,机械臂关节在各个位置的目标角度、目标角速度和目标角加速度转发至控制主站。

2. 根据权利要求1所述的机器人暂停运动控制方法,其特征在于,若所述当前运动为点到点运动或回到原点运动,

所述获取并记录当前运动的目标点和运动的剩余时间,以及读取机械臂末端的当前位置、当前速度和当前加速度,并根据所述当前位置、当前速度和当前加速度计算暂停运动轨迹的方法包括如下步骤:

计算当前点到点或回到原点运动的目标点,根据当前运动的总时长与当前运动已经历的时间计算当前运动的剩余时间,记录所述目标点和剩余时间;

读取机械臂当前的关节角度,关节角速度,以及关节角加速度;

依据最大加速度使速度降为0的原则,计算各个关节速度降为0的时间,记为暂停运行指令执行时间;

根据暂停运行指令执行时间得到暂停运行轨迹。

所述计算机械臂每个关节的目标角度、目标角速度和目标角加速度的方法包括如下步骤:

根据机械臂当前的关节角速度和所述暂停运行指令执行时间计算各个机械臂关节的目标角加速度;

根据所述目标角加速度计算各个机械臂关节的目标角速度与关节目标角度。

3. 根据权利要求2所述的机器人暂停运动控制方法,其特征在于,所述当前点到点或回到原点运动的目标点 θ_1 ,当前运动的剩余时间: $T_r = T_0 - t_0$,其中 T_0 是当前运动的总时长, t_0 是当前运动已经历的时间;

所述各个关节速度降为0的时间计算公式为:

$$T_i = \frac{\dot{\theta}_{i0}}{\ddot{\theta}_{imax}}$$

其中,i代表关节的标号,选取 T_i 中的最大值,记为暂停运行指令执行时间T;

机械臂关节的目标角加速度 $\ddot{\theta}_{imax}$ 计算公式为:

$$\ddot{\theta}_{imax} = \frac{\dot{\theta}_{i0}}{T}$$

各个机械臂关节的目标角速度 $\dot{\theta}_i$ 与关节目标角度 θ_i 的计算公式分别为:

$$\dot{\theta}_i = \dot{\theta}_{i0} - \ddot{\theta}_{imax} t$$

$$\theta_i = \dot{\theta}_{i0}t - \frac{\ddot{\theta}_{imax}t^2}{2}$$

式中, θ_0 为机械臂当前的关节角度, $\dot{\theta}_0$ 为关节角速度, $\ddot{\theta}_0$ 为关节角加速度; t表示暂停运动过程中的时刻。

4. 根据权利要求1所述的机器人暂停运动控制方法,其特征在于,若所述当前运动为直线运动;

所述获取并记录当前运动的目标点和运动的剩余时间,以及读取机械臂末端的当前位置、当前速度和当前加速度,并根据所述当前位置、当前速度和当前加速度计算暂停运动轨迹的方法包括:

计算当前直线运动的目标点,根据当前运动的总时长与当前运动已经历的时间计算当前运动的剩余时间,记录所述目标点和剩余时间;

根据直线运动轨迹计算机械臂当前时刻的轨迹,关节角速度和关节角加速度;

根据所述轨迹,关节角速度和关节角加速度计算暂停直线运动需要走过的轨迹,得到暂停运动轨迹,并记录恢复运动时机械臂沿直线已走过的轨迹;

所述计算机械臂每个关节的目标角度、目标角速度和目标角加速度的方法包括如下步骤:

计算暂停运动中机械臂任一时刻t的轨迹位置、轨迹速度和轨迹加速度;

根据所述轨迹位置、轨迹速度和轨迹加速度计算机械臂末端的目标位置、速度与加速度;

使用逆运动学方法根据所述目标位置,速度与加速度计算机械臂关节的目标角度、目标角速度和目标角加速度。

5. 根据权利要求4所述的机器人暂停运动控制方法,其特征在于,所述当前直线运动的目标点 s_e ,当前运动的剩余时间: $T_r = T_0 - t_0$,其中, T_0 是当前运动的总时长, t_0 是当前运动已经历的时间;

暂停直线运动需要走过的轨迹计算公式为:

$$s_1 = \frac{\dot{s}_0^2}{2a_{max}}$$

其中, a_{max} 为直线运动最大的加速度;记录 $s_2 = s_0 + s_1$,其中, s_2 为恢复运动时机械臂沿直线已走过的轨迹;

任一时刻t的轨迹位置、轨迹速度和轨迹加速度:

$$s = \dot{s}_0t - \frac{a_{max}t^2}{2}$$

$$\dot{s}(t) = \dot{s}_0 - a_{max}t$$

$$\ddot{s}(t) = a_{max}$$

其中, s_0 为机械臂当前时刻的轨迹, \dot{s}_0 为关节角速度, \ddot{s}_0 为关节角加速度, s为轨迹位置, $\dot{s}(t)$ 为轨迹速度, $\ddot{s}(t)$ 为轨迹加速度;

机械臂末端的目标位置、速度与加速度的计算公式为:

$$\mathbf{P}_t = s_t \vec{\mathbf{r}}, \dot{\mathbf{P}}_t = \dot{s}_t \vec{\mathbf{r}}, \ddot{\mathbf{P}}_t = \ddot{s}_t \vec{\mathbf{r}}$$

其中, P_t 为目標位置, \dot{P}_t 为速度, \ddot{P}_t 为加速度, \bar{r} 是当前直线运动的单位向量;

所述机械臂关节的目标角度、目标角速度和目标角加速度的计算公式为:

$$\theta_t = \text{invKinematics}(P_t)$$

$$\dot{\theta}_t = J^+ \dot{P}_t$$

$$\ddot{\theta}_t = J^+ (\ddot{P}_t - J \dot{\theta}_t)$$

其中, θ_t 为目標角度, $\dot{\theta}_t$ 为目標角速度, $\ddot{\theta}_t$ 为目標角加速度, $\text{invKinematics}()$ 表示逆运动学的计算公式, J 为雅可比矩阵, J^+ 为雅可比矩阵的一阶导, J^+ 为的伪逆。

6. 根据权利要求1所述的机器人暂停运动控制方法, 其特征在于, 若所述当前运动为圆弧运动;

所述获取并记录当前运动的目标点和运动的剩余时间, 以及读取机械臂末端的当前位置、当前速度和当前加速度, 并根据所述当前位置、当前速度和当前加速度计算暂停运动轨迹的方法包括:

计算当前圆弧运动的目标点, 根据当前运动的总时长与当前运动已经历的时间计算当前运动的剩余时间, 记录所述目标点和剩余时间;

根据圆弧运动轨迹计算机械臂当前时刻的轨迹, 关节角速度和关节角加速度;

根据所述轨迹, 关节角速度和关节角加速度计算暂停圆弧运动需要走过的轨迹, 得到暂停运行轨迹, 并记录恢复运动时机械臂沿圆弧已走过的轨迹;

所述计算机械臂每个关节的目标角度、目标角速度和目标角加速度的方法包括如下步骤:

计算暂停运动中机械臂任一时刻 t 的轨迹位置、轨迹速度和轨迹加速度;

根据所述轨迹位置、轨迹速度和轨迹加速度计算机械臂末端的目标位置、速度与加速度;

使用逆运动学方法根据所述目标位置, 速度与加速度计算机械臂关节的目标角度、目标角速度和目标角加速度。

7. 根据权利要求6所述的机器人暂停运动控制方法, 其特征在于, 所述当前圆弧运动的目标点 s_e , 当前运动的剩余时间: $T_r = T_0 - t_0$, 其中, T_0 是当前运动的总时长, t_0 是当前运动已经历的时间;

暂停圆弧运动需要走过的轨迹计算公式为:

$$s_t = \frac{\dot{s}_0^2}{2a_{\max}}$$

其中, a_{\max} 为圆弧运动最大的加速度; 记录 $s_2 = s_0 + s_1$, 其中, s_2 为恢复运动时机械臂沿圆弧已走过的轨迹;

任一时刻 t 的轨迹位置、轨迹速度和轨迹加速度:

$$s = \dot{s}_0 t - \frac{a_{\max} t^2}{2}$$

$$\dot{s}(t) = \dot{s}_0 - a_{\max} t$$

$$\ddot{s}(t) = a_{\max}$$

其中, s_0 为机械臂当前时刻的轨迹, \dot{s}_0 为关节角速度, \ddot{s}_0 为关节角加速度, s 为轨迹位置, $\dot{s}(t)$ 为轨迹速度, $\ddot{s}(t)$ 为轨迹加速度;

机械臂末端的目标位置 $P_t = [x_t, y_t, z_t]^T$:

$$\begin{cases} x_t = x_c + R \cos(\theta_i) a_1 + R \sin(\theta_i) b_1 \\ y_t = y_c + R \cos(\theta_i) a_2 + R \sin(\theta_i) b_2 \\ z_t = z_c + R \cos(\theta_i) a_3 + R \sin(\theta_i) b_3 \end{cases}$$

其中, θ_t 为圆弧 $P_0 P_t$ 对应的圆心角, 满足 $\theta_t = \frac{s_t}{R}$, $a_1, a_2, a_3, b_1, b_2, b_3, R$ 等参数均是当前圆弧运动轨迹已知参数;

机械臂末端的目标速度 \dot{P}_t 和加速度 \ddot{P}_t 计算公式为:

$$\dot{P}_t = \dot{s}_t \bar{r}, \ddot{P}_t = \ddot{s}_t \bar{r}$$

其中, \bar{r} 是点 P_t 的切向量;

所述机械臂关节的目标角度、目标角速度和目标角加速度的计算公式为:

$$\theta_t = \text{invKinematics}(P_t)$$

$$\dot{\theta}_t = \mathbf{J}^+ \dot{P}_t$$

$$\ddot{\theta}_t = \mathbf{J}^+ (\ddot{P}_t - \mathbf{J} \dot{\theta}_t)$$

其中, θ_t 为目标角度, $\dot{\theta}_t$ 为目标角速度, $\ddot{\theta}_t$ 为目标角加速度, $\text{invKinematics}()$ 表示逆运动学的计算公式, \mathbf{J} 为雅可比矩阵, \mathbf{J}^+ 为雅可比矩阵的一阶导, \mathbf{J}^+ 为的伪逆。

8. 根据权利要求1至7任一项所述的机器人暂停运动控制方法, 其特征在于, 还包括:

在执行该暂停运动指令后, 异步触发暂停运动执行函数并通过第一接口调用暂停运动规划函数; 其中, 所述第一接口是基于ROS上创建OROCOS的实时输入/输出接口;

执行所述暂停运动规划函数, 并计算暂停运动轨迹;

以及通过第二接口将所述机械臂关节的目标角度、目标角速度和目标角加速度发送给设备通信软件转发至控制主站; 其中, 所述第二接口是基于ROS上创建OROCOS的实时输入/输出接口。

9. 根据权利要求8所述的机器人暂停运动控制方法, 其特征在于, 计算机械臂关节的目标角度、目标角速度和目标角加速度的方法包括:

在机械臂开始在所述暂停运动轨迹上运动后, 统计机械臂的运动时间 t_n ;

若所述运动时间满足: $t_n \leq T$, 每隔设定时间 τ 计算一次机械臂每个关节的目标角度、目标角速度和目标角加速度; 若运动时间 t_n 满足: $t_n > T$, 则停止运动;

其中, T 为暂停运行指令执行时间。

10. 一种机器人暂停运动控制系统, 其特征在于, 包括: 总控模块、算法模块和通信管理模块;

所述总控模块, 用于接收控制端传送的暂停运动指令; 在当前机器人处于点到点运动、直线运动、圆弧运动或回到原点运动的任意一种当前运动状态下, 执行所述暂停运动指令;

所述算法模块, 用于获取当前运动的目标点和运动的剩余时间, 将所述目标点和运动的剩余时间保存为恢复当前运动的轨迹计算参数; 读取机械臂末端的当前位置、当前速度

和当前加速度，并计算机械臂末端由当前位置、当前速度和当前加速度变化到停止时所经过的暂停运动轨迹；计算机械臂末端在所述暂停运动轨迹上运行时，机械臂关节在各个位置的目标角度、目标角速度和目标角加速度；

所述通信管理模块，用于将所述目标角度、目标角速度和目标角加速度转发至控制主站。

机器人暂停运动控制方法和系统

技术领域

[0001] 本发明涉及机器人控制技术领域,特别是涉及一种机器人暂停运动控制方法和系统。

背景技术

[0002] Robot Operating System (ROS) 是开源的机器人操作系统,可以为机器人开发者提供一个标准化的、开源的编程框架。但是ROS目前不支持实时线程操作。Open Robot Control Software (OROCOS) 也是一种开源的机器人控制软件编程框架,它的特点是支持实时的线程操作,但是它的开放性,通用性没有ROS好。

[0003] 因此,机器人暂停运动是机器人一种重要运动方式,目前在一些应用方案中,在ROS上采用OROCOS,但现有技术方案在搭建的架构上,不能构成一个完整的机器人控制器软件,在执行暂停运动时,无法充分利用ROS和OROCOS的特性,系统开发成本高,控制效果差。

发明内容

[0004] 基于此,有必要针对上述技术问题,提供一种机器人暂停运动控制方法,降低系统开发成本,提高控制效果。

[0005] 一种机器人暂停运动控制方法,包括:

[0006] 接收控制端传送的暂停运动指令;

[0007] 在当前机器人处于点到点运动、直线运动、圆弧运动或回到原点运动的任意一种当前运动状态下,执行所述暂停运动指令;

[0008] 获取当前运动的目标点和运动的剩余时间,将所述目标点和运动的剩余时间保存为恢复当前运动的轨迹计算参数;读取机械臂末端的当前位置、当前速度和当前加速度,并计算机械臂末端由当前位置、当前速度和当前加速度变化到停止时所经过的暂停运动轨迹;

[0009] 计算机械臂末端在所述暂停运动轨迹上运行时,机械臂关节在各个位置的目标角度、目标角速度和目标角加速度转发至控制主站。

[0010] 一种机器人暂停运动控制系统,包括:总控模块、算法模块和通信管理模块;

[0011] 所述总控模块,用于接收控制端传送的暂停运动指令;在当前机器人处于点到点运动、直线运动、圆弧运动或回到原点运动的任意一种当前运动状态下,执行所述暂停运动指令;

[0012] 所述算法模块,用于获取当前运动的目标点和运动的剩余时间,将所述目标点和运动的剩余时间保存为恢复当前运动的轨迹计算参数;读取机械臂末端的当前位置、当前速度和当前加速度,并计算机械臂末端由当前位置、当前速度和当前加速度变化到停止时所经过的暂停运动轨迹;计算机械臂末端在所述暂停运动轨迹上运行时,机械臂关节在各个位置的目标角度、目标角速度和目标角加速度;

[0013] 所述通信管理模块,用于将所述目标角度、目标角速度和目标角加速度转发至控

制主站。

[0014] 上述机器人暂停运动控制方法和系统,在接收到控制端传送的暂停运动指令后,在当前机器人处于点到点运动、直线运动、圆弧运动或回到原点运动的任意一种当前运动状态下,计算暂停运动的运动轨迹,并实时计算机械臂各个关节在该运动轨迹上运行的目标角度、目标角速度和目标角加速度等状态参数,并转发至控制主站;实现对机器人的暂停运动控制过程;该技术方案可以构成一个完整的机器人暂停运动控制系统,降低机器人控制系统开发成本,提高控制效果。

附图说明

- [0015] 图1是本发明实施例的机器人暂停运动控制方法流程图;
- [0016] 图2是机器人暂停运动控制系统结构示意图;
- [0017] 图3是总控模块执行算法流程图;
- [0018] 图4是算法模块执行算法流程图;
- [0019] 图5是一应用实例的机器人控制系统的硬件结构模型;
- [0020] 图6是基于ROS和OROCOS的搭建的软件架构图;
- [0021] 图7是控制器状态机的状态变化示意图;
- [0022] 图8是设备状态机的状态变化示意图。

具体实施方式

- [0023] 下面结合附图阐述本发明的机器人暂停运动控制方法的实施例。
- [0024] 本发明实施例中,所述暂停运动,是指机械臂暂停当前运动指令,速度降为0,当收到恢复指令后,将沿当前规划的轨迹继续运动,直到达到当前指令的目标位置。
- [0025] 参考图1所示,图1是本发明实施例的机器人暂停运动控制方法流程图,包括:
- [0026] S10,接收控制端传送的暂停运动指令;
- [0027] 上述步骤中,可以利用预设的通信协议并以异步远程过程调用的方式接收暂停运动指令;
- [0028] 在此过程中,可以是操作者通过控制端的人机交互界面生成暂停运动指令,该指令无需传递参数;通过预设的通信协议,如基于IEC(The Internet Communications Engine,互联网通信引擎)开发的通信协议,以异步远程过程调用的方式从人机交互界面接收暂停运动指令。
- [0029] S20,在当前机器人处于点到点运动、直线运动、圆弧运动或回到原点运动的任意一种当前运动状态下,执行所述暂停运动指令;
- [0030] 本步骤中,通过判断当前机器人运动状态是否为可执行暂停运动的点到点运动、直线运动、圆弧运动或回到原点运动状态,若是,执行该暂停运动指令;
- [0031] 在一个实施例中,可以实时监测控制器状态机的状态,当转换为执行暂停运动状态时,调用暂停运动规划函数,开始执行急停运动规划流程;
- [0032] 对于判断当前机器人运动状态是否为可执行暂停运动的指定运动状态,可以通过判断控制器状态机是否为点到点运动、直线运动、圆弧运动、回到原点运动的四种状态之一,若是,执行调用所述暂停运动规划函数的步骤,否则,拒绝执行该暂停运动指令。

[0033] S30, 获取当前运动的目标点和运动的剩余时间, 将所述目标点和运动的剩余时间保存为恢复当前运动的轨迹计算参数; 读取机械臂末端的当前位置、当前速度和当前加速度, 并计算机械臂末端由当前位置、当前速度和当前加速度变化到停止时所经过的暂停运动轨迹;

[0034] 在一个实施例中, 执行所述暂停运动指令后, 可以异步触发暂停运动执行函数并通过第一接口调用暂停运动规划函数; 其中, 所述第一接口是基于ROS上创建OROCOS的实时输入/输出接口;

[0035] 进一步的, 可以通过OROCOS的Operational Caller方法调用所述暂停运动规划函数;

[0036] 执行所述暂停运动规划函数, 获取并记录当前运动的目标点和运动的剩余时间, 以及读取机械臂末端的当前位置、当前速度和当前加速度, 并根据所述当前位置、当前速度和当前加速度计算暂停运动轨迹;

[0037] 在一个实施例中, 执行所述暂停运动规划函数, 将所述控制器状态机转换为执行暂停运动(Active.Halt)状态。

[0038] 作为实施例, 若所述当前运动为点到点运动或回到原点运动, 则计算暂停运动轨迹的方法可以包括如下步骤:

[0039] (A1) 计算当前点到点或回到原点运动的目标点, 根据当前运动的总时长与当前运动已经历的时间计算当前运动的剩余时间, 记录所述目标点和剩余时间;

[0040] 所述当前点到点或回到原点运动的目标点 θ_1 , 当前运动的剩余时间: $T_r = T_0 - t_0$, 其中 T_0 是当前运动的总时长, t_0 是当前运动已经历的时间;

[0041] (A2) 读取机械臂当前的关节角度, 关节角速度, 以及关节角加速度;

[0042] 具体的, 读取机械臂当前的关节角度 θ_0 , 关节角速度 $\dot{\theta}_0$, 关节角加速度 $\ddot{\theta}_0$ 。

[0043] (A3) 将当前时间记为0, 依据最大加速度使速度降为0的原则, 计算各个关节速度降为0的时间, 记为暂停运行指令执行时间;

[0044] 所述各个关节速度降为0的时间计算公式为:

$$[0045] T_i = \frac{\dot{\theta}_{i0}}{\ddot{\theta}_{imax}}$$

[0046] 其中, i代表关节的标号, 选取 T_i 中的最大值, 记为暂停运行指令执行时间T;

[0047] (A4) 根据暂停运行指令执行时间T得到暂停运行轨迹。

[0048] 在一个实施例中, 若所述当前运动为直线运动, 则计算暂停运动轨迹的方法可以包括如下步骤:

[0049] (B1) 计算当前直线运动的目标点, 根据当前运动的总时长与当前运动已经历的时间计算当前运动的剩余时间, 记录所述目标点和剩余时间;

[0050] 具体的, 所述当前直线运动的目标点 s_e , 当前运动的剩余时间: $T_r = T_0 - t_0$, 其中, T_0 是当前运动的总时长, t_0 是当前运动已经历的时间。

[0051] (B2) 根据直线运动轨迹计算机械臂当前时刻的轨迹, 关节角速度和关节角加速度;

[0052] 具体的, 当前直线运动轨迹已由已知算法规划, 因此可以计算机械臂当前时刻的

轨迹 s_0 ,关节角速度 $\dot{\theta}_0$,关节角加速度 $\ddot{\theta}_0$ 。

[0053] (B3) 根据所述轨迹,关节角速度和关节角加速度计算暂停直线运动需要走过的轨迹,得到暂停运动轨迹,并记录恢复运动时机械臂沿直线已走过的轨迹;

[0054] 具体的,暂停直线运动需要走过的轨迹计算公式为:

$$[0055] s_1 = \frac{\dot{\theta}_0^2}{2a_{\max}}$$

[0056] 其中, a_{\max} 为直线运动最大的加速度;记录 $s_2=s_0+s_1$,其中, s_2 为恢复运动时机械臂沿直线已走过的轨迹。

[0057] 在一个实施例中,若所述当前运动为圆弧运动,则计算暂停运动轨迹的方法可以包括如下步骤:

[0058] (C1) 计算当前圆弧运动的目标点,根据当前运动的总时长与当前运动已经历的时间计算当前运动的剩余时间,记录所述目标点和剩余时间;

[0059] 所述当前圆弧运动的目标点 s_e ,当前运动的剩余时间: $T_r=T_0-t_0$,其中, T_0 是当前运动的总时长, t_0 是当前运动已经历的时间;

[0060] (C2) 根据圆弧运动轨迹计算机械臂当前时刻的轨迹,关节角速度和关节角加速度;

[0061] 具体的,当前圆弧运动轨迹已由已知算法规划,因此可以计算机械臂当前时刻的轨迹 s_0 ,关节角速度 $\dot{\theta}_0$,关节角加速度 $\ddot{\theta}_0$ 。

[0062] (C3) 根据所述轨迹,关节角速度和关节角加速度计算暂停圆弧运动需要走过的轨迹,得到暂停运行轨迹,并记录恢复运动时机械臂沿圆弧已走过的轨迹;

[0063] 暂停圆弧运动需要走过的轨迹计算公式为:

$$[0064] s_1 = \frac{\dot{\theta}_0^2}{2a_{\max}}$$

[0065] 其中, a_{\max} 为圆弧运动最大的加速度;记录 $s_2=s_0+s_1$,其中, s_2 为恢复运动时机械臂沿圆弧已走过的轨迹。

[0066] S40,计算机械臂末端在所述暂停运动轨迹上运行时,机械臂关节在各个位置的目标角度、目标角速度和目标角加速度转发至控制主站。

[0067] 在一个实施例中,通过第二接口将所述机械臂关节的目标角度、目标角速度和目标角加速度发送给设备通信软件转发至控制主站;其中,所述第二接口是基于ROS上创建OROCOS的实时输入/输出接口。

[0068] 作为实施例,对于计算机械臂末端在所述暂停运动轨迹上运行时,机械臂关节在各个位置的机械臂关节的目标角度、目标角速度和目标角加速度的方法,可以包括如下步骤:

[0069] (a) 在机械臂开始在所述暂停运动轨迹上运动后,统计机械臂的运动时间 t_n ;

[0070] (b) 若所述运动时间满足: $t_n \leq T$,每隔设定时间 τ 计算一次机械臂每个关节的目标角度、目标角速度和目标角加速度; $\tau=1/f$, f 表示刷新频率;一般情况下,所述 τ 为1毫秒;

[0071] (c) 若运动时间 t_n 满足: $t_n > T$,则停止运动;其中, T 为暂停运行指令执行时间。

[0072] 如上述实施例,若所述当前运动为点到点运动或回到原点运动,则计算机械臂每

个关节的目标角度、目标角速度和目标角加速度的方法包括如下步骤：

[0073] (A5) 根据机械臂当前的关节角速度和所述暂停运行指令执行时间计算各个机械臂关节的目标角加速度；

[0074] 具体的，机械臂关节的目标角加速度 $\ddot{\theta}_{i_{\max}}$ 计算公式为：

$$[0075] \ddot{\theta}_{i_{\max}} = \frac{\dot{\theta}_{i_0}}{T}$$

[0076] (A6) 根据所述目标角加速度计算各个机械臂关节的目标角速度与关节目标角度；

[0077] 具体的，各个机械臂关节的目标角速度 $\dot{\theta}_i$ 与关节目标角度 θ_i 的计算公式分别为：

$$[0078] \dot{\theta}_i = \dot{\theta}_{i_0} - \ddot{\theta}_{i_{\max}} t$$

$$[0079] \theta_i = \theta_{i_0} + \dot{\theta}_{i_0} t - \frac{\ddot{\theta}_{i_{\max}} t^2}{2}$$

[0080] 式中， θ_0 为机械臂当前的关节角度， $\dot{\theta}_0$ 为关节角速度， $\ddot{\theta}_0$ 为关节角加速度；t 表示暂停运动过程中的时刻。

[0081] 如上述实施例，若所述当前运动为直线运动，则计算机械臂每个关节的目标角度、目标角速度和目标角加速度的方法包括如下步骤：

[0082] (B4) 计算暂停运动中机械臂任一时刻 t 的轨迹位置、轨迹速度和轨迹加速度；

[0083] 具体的，任一时刻 t 的轨迹位置、轨迹速度和轨迹加速度：

$$[0084] s = s_0 t - \frac{a_{\max} t^2}{2}$$

$$[0085] \dot{s}(t) = \dot{s}_0 - a_{\max} t$$

$$[0086] \ddot{s}(t) = a_{\max}$$

[0087] 其中， s_0 为机械臂当前时刻的轨迹， \dot{s}_0 为关节角速度， \ddot{s}_0 为关节角加速度，s 为轨迹位置， $\dot{s}(t)$ 为轨迹速度， $\ddot{s}(t)$ 为轨迹加速度；

[0088] (B5) 根据所述轨迹位置、轨迹速度和轨迹加速度计算机械臂末端的目标位置、速度与加速度；

[0089] 具体的，机械臂末端的目标位置、速度与加速度的计算公式为：

$$[0090] \mathbf{P}_t = s_t \bar{\mathbf{r}}, \dot{\mathbf{P}}_t = \dot{s}_t \bar{\mathbf{r}}, \ddot{\mathbf{P}}_t = \ddot{s}_t \bar{\mathbf{r}}$$

[0091] 其中， P_t 为目标位置， $\dot{\mathbf{P}}_t$ 为速度， $\ddot{\mathbf{P}}_t$ 为加速度， $\bar{\mathbf{r}}$ 是当前直线运动的单位向量；

[0092] (B6) 使用逆运动学方法根据所述目标位置，速度与加速度计算机械臂关节的目标角度、目标角速度和目标角加速度；

[0093] 具体的，所述机械臂关节的目标角度、目标角速度和目标角加速度的计算公式为：

$$[0094] \theta_t = \text{invKinematics}(P_t)$$

$$[0095] \dot{\theta}_t = \mathbf{J}^+ \dot{\mathbf{P}}_t$$

$$[0096] \ddot{\theta}_t = \mathbf{J}^+ (\ddot{\mathbf{P}}_t - \mathbf{J} \dot{\theta}_t)$$

[0097] 其中， θ_t 为目标角度， $\dot{\theta}_t$ 为目标角速度， $\ddot{\theta}_t$ 为目标角加速度， $\text{invKinematics}()$ 表示逆运动学的计算公式， \mathbf{J} 为雅可比矩阵， \mathbf{J}^+ 为雅可比矩阵的一阶导， \mathbf{J}^+ 为的伪逆。

[0098] 如上述实施例,若所述当前运动为圆弧运动,则计算机械臂每个关节的目标角度、目标角速度和目标角加速度的方法包括如下步骤:

[0099] (C4) 计算暂停运动中机械臂任一时刻t的轨迹位置、轨迹速度和轨迹加速度;

[0100] 具体的,任一时刻t的轨迹位置、轨迹速度和轨迹加速度:

$$[0101] s = \dot{s}_0 t - \frac{a_{\max} t^2}{2}$$

$$[0102] \dot{s}(t) = \dot{s}_0 - a_{\max} t$$

$$[0103] \ddot{s}(t) = a_{\max}$$

[0104] 其中, s_0 为机械臂当前时刻的轨迹, \dot{s}_0 为关节角速度, \ddot{s}_0 为关节角加速度, s 为轨迹位置, $\dot{s}(t)$ 为轨迹速度, $\ddot{s}(t)$ 为轨迹加速度;

[0105] (C5) 根据所述轨迹位置、轨迹速度和轨迹加速度计算机械臂末端的目标位置、速度与加速度;

[0106] 具体的,机械臂末端的目标位置 $P_t = [x_t, y_t, z_t]^T$:

$$[0107] \begin{cases} x_t = x_c + R \cos(\theta_t) a_1 + R \sin(\theta_t) b_1 \\ y_t = y_c + R \cos(\theta_t) a_2 + R \sin(\theta_t) b_2 \\ z_t = z_c + R \cos(\theta_t) a_3 + R \sin(\theta_t) b_3 \end{cases}$$

[0108] 其中, θ_t 为圆弧 $P_0 P_t$ 对应的圆心角, 满足 $\theta_t = \frac{s_t}{R}$, $a_1, a_2, a_3, b_1, b_2, b_3, R$ 等参数均是当前圆弧运动轨迹已知参数;

[0109] 机械臂末端的目标速度 \dot{P}_t 和加速度 \ddot{P}_t 计算公式为:

$$[0110] \dot{P}_t = \dot{s}_t \bar{r}, \ddot{P}_t = \ddot{s}_t \bar{r}$$

[0111] 其中, \bar{r} 是点 P_t 的切向量;

[0112] (C6) 使用逆运动学方法根据所述目标位置,速度与加速度计算机械臂关节的目标角度、目标角速度和目标角加速度。

[0113] 所述机械臂关节的目标角度、目标角速度和目标角加速度的计算公式为:

$$[0114] \theta_t = \text{invKinematics}(P_t)$$

$$[0115] \dot{\theta}_t = \mathbf{J}^+ \dot{P}_t$$

$$[0116] \ddot{\theta}_t = \mathbf{J}^+ (\ddot{P}_t - \mathbf{J} \dot{\theta}_t)$$

[0117] 其中, θ_t 为目标角度, $\dot{\theta}_t$ 为目标角速度, $\ddot{\theta}_t$ 为目标角加速度, $\text{invKinematics}()$ 表示逆运动学的计算公式, \mathbf{J} 为雅可比矩阵, \mathbf{J}^+ 为雅可比矩阵的一阶导, \mathbf{J}^+ 为的伪逆。

[0118] 上述实施例的技术方案,利用OROCOS的实时输入/输出接口,通过设定通信协议以异步远程过程调用的方式接收暂停运动指令,启动暂停运动指令触发函数,调用机器人运动规划的暂停运动执行函数,根据该函数计算暂停运动的运动轨迹,并实时计算机械臂各个关节在该运动轨迹上运行的目标角度、目标角速度和目标角加速度等状态参数,通过设备通信软件将上述状态参数转发至控制主站,实现对机器人的暂停运动控制过程;该技术方案可以进行控制指令的接收、解析,算法调用、执行,参数实时计算和传输等功能,可以构成一个完整的机器人暂停运动控制系统,降低机器人控制系统开发成本,提高控制效果;另

外结合了控制器状态机的应用,实现对算法处理过程的优化控制,进一步提高了控制效果。

[0119] 针对于机器人暂停运动控制方法,本发明提供了该控制方法对应的机器人暂停运动控制系统,

[0120] 参考图2所示,图2是机器人暂停运动控制系统结构示意图,包括:总控模块、算法模块和通信管理模块;

[0121] 所述总控模块,用于接收控制端传送的暂停运动指令;在当前机器人处于点到点运动、直线运动、圆弧运动或回到原点运动的任意一种当前运动状态下,执行所述暂停运动指令;

[0122] 所述算法模块,用于获取当前运动的目标点和运动的剩余时间,将所述目标点和运动的剩余时间保存为恢复当前运动的轨迹计算参数;读取机械臂末端的当前位置、当前速度和当前加速度,并计算机械臂末端由当前位置、当前速度和当前加速度变化到停止时所经过的暂停运动轨迹;计算机械臂末端在所述暂停运动轨迹上运行时,机械臂关节在各个位置的目标角度、目标角速度和目标角加速度;

[0123] 所述通信管理模块,用于将所述目标角度、目标角速度和目标角加速度转发至控制主站。

[0124] 在一个实施例中,所述总控模块通过第一接口与算法模块进行通信连接,算法模块通过第二接口与通信管理模块进行通信连接,所述第一接口、第二接口是基于ROS上创建OROCOS的实时输入/输出接口;

[0125] 所述总控模块执行暂停运动指令后,异步触发暂停运动执行函数并通过第一接口调用暂停运动规划函数;

[0126] 所述算法模块执行所述暂停运动规划函数,并计算暂停运动轨迹;

[0127] 以及通过第二接口将所述机械臂关节的目标角度、目标角速度和目标角加速度发送给通信管理模块。

[0128] 上述实施例的技术方案,利用OROCOS的实时输入/输出接口,总控模块通过设定通信协议以异步远程过程调用的方式接收暂停运动指令,启动暂停运动指令触发函数,调用算法模块的机器人运动规划的暂停运动执行函数,根据该函数计算暂停运动的运动轨迹,并实时计算机械臂各个关节在该运动轨迹上运行的目标角度、目标角速度和目标角加速度等状态参数,通信管理模块将上述状态参数转发至控制主站,实现对机器人的暂停运动控制过程;该技术方案可以进行控制指令的接收、解析,算法调用、执行,参数实时计算和传输等功能,可以构成一个完整的机器人暂停运动控制系统,降低机器人控制系统开发成本,提高控制效果。

[0129] 作为实施例,总控模块执行算法流程,可以参考图3所示,图3是总控模块执行算法流程图;具体如下:

[0130] 1) 操作者通过人机交互界面生成暂停运动指令。

[0131] 2) 总控模块通过通信协议,以异步远程过程调用的方式从人机交互界面接收暂停运动指令。

[0132] 3) 指令到达总控模块后,异步触发暂停运动执行函数,该函数首先判断控制器状态机是否为点到点运动,直线运动,圆弧运动,回到原点的四种状态之一。若不是,则拒绝执行此次暂停运行指令。

[0133] 4) 若控制器状态机是点到点运动, 直线运动, 圆弧运动, 回到原点的四种状态之一, 则执行如下操作:

[0134] a) 通过OROCOS的Operational Caller方法调用算法模块的暂停运动规划函数。

[0135] b) 将控制器状态机转换为执行暂停运动(Active.Halt)状态。

[0136] 作为实施例, 算法模块执行算法流程, 可以参考图4所示, 图4是算法模块执行算法流程图; 具体如下:

[0137] 1) 暂停运动规划函数被总控模块调用, 开始执行暂停运动规划流程。

[0138] 2) 若当前运动是回零运动或者点到点运动, 则调用点到点暂停规划算法:

[0139] a) 记录当前运动的目标点 θ_1 ;

[0140] b) 记录当前运动的剩余时间: $T_r = T - t$ 。其中T是当前运动的总时长, t是当前运动已经历的时间;

[0141] c) 读取机械臂当前的关节角度 θ_0 , 关节角速度 $\dot{\theta}_0$, 关节角加速度 $\ddot{\theta}_0$;

[0142] d) 将当前时间记为0, 依据最大加速度使速度降为0的原则, 计算各个关节速度降为0的时间:

$$[0143] T_i = \frac{\dot{\theta}_{i0}}{\ddot{\theta}_{imax}}$$

[0144] 其中, i代表关节的标号。选取 T_i 中的最大值, 记为暂停指令执行时间, 记为T。

[0145] e) 计算各个关节的加速度为:

$$[0146] \ddot{\theta}_{imax} = \frac{\dot{\theta}_{i0}}{T} \quad (1)$$

[0147] f) t时刻对应的速度与关节角位置分别为:

$$[0148] \dot{\theta}_i = \dot{\theta}_{i0} - \ddot{\theta}_{imax} t \quad (2)$$

$$[0149] \theta_i = \dot{\theta}_{i0} t - \frac{\ddot{\theta}_{imax} t^2}{2} \quad (3)$$

[0150] 3) 若当前运动是直线运动, 则调用直线暂停的算法:

[0151] a) 记录当前运动的目标点 s_e 。

[0152] b) 记录当前运动的剩余时间: $T_r = T - t$ 。其中T是当前运动的总时长, t是当前运动已经历的时间;

[0153] c) 由于当前直线运动轨迹可以由五次多项式算法规划, 因此可以计算机械臂当前时刻的轨迹 s_0 , 关节角速度 \dot{s}_0 , 关节角加速度 \ddot{s}_0 。

[0154] d) 计算暂停直线运动需要走过的轨迹:

$$[0155] s_1 = \frac{\dot{s}_0^2}{2a_{max}}$$

[0156] 其中, a_{max} 就直线运动最大的加速度。

[0157] e) 记录恢复运动时, 机械臂沿直线已走过的轨迹: $s_2 = s_0 + s_1$;

[0158] f) 计算暂停运动中机械臂任一时刻t的轨迹, 轨迹速度, 轨迹加速度:

$$[0159] s = \dot{s}_0 t - \frac{a_{max} t^2}{2} \quad (4)$$

[0160] $\dot{s}(t) = \dot{s}_0 - a_{\max} t$ (5)

[0161] $\ddot{s}(t) = a_{\max}$ (6)

[0162] g) 计算机械臂末端的目标位置,速度与加速度:

[0163] $\mathbf{P}_t = s_t \bar{\mathbf{r}}, \dot{\mathbf{P}}_t = \dot{s}_t \bar{\mathbf{r}}, \ddot{\mathbf{P}}_t = \ddot{s}_t \bar{\mathbf{r}}$ (7)

[0164] 其中, $\bar{\mathbf{r}}$ 是直线运动的单位向量。

[0165] h) 使用逆运动学方法,计算关节角的位置:

[0166] $\theta_t = \text{invKinematics}(\mathbf{P}_t)$ (8)

[0167] i) 关节的角速度与角加速度,可由如下表达式求出:

[0168] $\dot{\theta}_t = \mathbf{J}^+ \dot{\mathbf{P}}_t$ (9)

[0169] $\ddot{\theta}_t = \mathbf{J}^+ (\ddot{\mathbf{p}}_t - \mathbf{J} \dot{\theta}_t)$ (10)

[0170] 其中, \mathbf{J} 为雅可比 (Jacobian) 矩阵, \mathbf{J}^+ 为的伪逆 (Pseudoinverse)。

[0171] 4) 若当前运动是圆弧运动,则调用圆弧暂停的算法:

[0172] a) 记录当前运动的目标轨迹 s_e ;

[0173] b) 记录当前运动的剩余时间: $T_r = T - t$ 。其中 T 是当前运动的总时长, t 是当前运动已经历的时间;

[0174] c) 由于当前圆弧运动轨迹可以由五次多项式算法规划,因此可以计算机械臂当前时刻的轨迹 s_0 , 关节角速度 \dot{s}_0 , 关节角加速度 \ddot{s}_0 ;

[0175] d) 计算暂停圆弧运动需要走过的轨迹:

[0176] $s_1 = \frac{\dot{s}_0^2}{2a_{\max}}$

[0177] 其中, a_{\max} 就直线运动最大的加速度。

[0178] e) 记录恢复运动时,机械臂沿圆弧已走过的轨迹: $s_2 = s_0 + s_1$

[0179] f) 计算暂停运动中机械臂任一时刻 t 的轨迹, 轨迹速度, 轨迹加速度:

[0180] $s = \dot{s}_0 t - \frac{a_{\max} t^2}{2}$ (11)

[0181] $\dot{s}(t) = \dot{s}_0 - a_{\max} t$ (12)

[0182] $\ddot{s}(t) = a_{\max}$ (13)

[0183] g) 计算末端的目标位置 $\mathbf{P}_t = [x_t, y_t, z_t]^T$:

$$\begin{cases} x_t = x_c + R \cos(\theta_t) a_1 + R \sin(\theta_t) b_1 \\ y_t = y_c + R \cos(\theta_t) a_2 + R \sin(\theta_t) b_2 \\ z_t = z_c + R \cos(\theta_t) a_3 + R \sin(\theta_t) b_3 \end{cases} \quad (14)$$

[0185] 其中, θ_t 为圆弧 $P_0 P_t$ 对应的圆心角, 满足 $\theta_t = \frac{s_t}{R}$, $a_1, a_2, a_3, b_1, b_2, b_3, R$ 等参数可以由圆弧运动五次多项式规划算法求得;

[0186] h) 求得末端的目标速度 $\dot{\mathbf{P}}_t$ 和加速度 $\ddot{\mathbf{P}}_t$:

[0187] $\dot{\mathbf{P}}_t = \dot{s}_t \dot{\mathbf{r}}, \ddot{\mathbf{P}}_t = \ddot{s}_t \dot{\mathbf{r}}$ (15)

[0188] 其中, $\dot{\mathbf{r}}$ 是点 P_t 的切向量;

[0189] i) 使用逆运动学方法, 计算关节角的位置:

[0190] $\theta_t = \text{invKinematics}(P_t)$ (16)

[0191] 其中, $\text{invKinematics}()$ 代表逆运动学的计算公式。

[0192] j) 关节的角速度与角加速度, 可由如下表达式求出:

[0193] $\dot{\theta}_t = \mathbf{J}^+ \dot{\mathbf{P}}_t$ (17)

[0194] $\ddot{\theta}_t = \mathbf{J}^+ (\ddot{\mathbf{p}}_t - \mathbf{J} \dot{\theta}_t)$ (18)

[0195] 其中, \mathbf{J} 为雅可比 (Jacobian) 矩阵, \mathbf{J} 为雅可比矩阵的一阶导, \mathbf{J}^+ 为的伪逆 (Pseudoinverse)。

[0196] 5) 运动时间 t_n 不超过运动时长 T :

[0197] a) 每隔1毫秒分别计算一次机械臂每个关节的目标角度, 目标角速度和目标角加速度;

[0198] b) 将机械臂每个关节的目标角度, 目标角速度和目标角加速度, 发送给通信管理模块。

[0199] 6) 运动时间 t_n 超过运动时长 T , 则停止运动。

[0200] 作为实施例, 通信管理模块, 可以进一步用于执行如下算法:

[0201] 读取机器人电机的状态信息, 根据机器人运动模型计算机器人机械臂的状态信息, 将机器人的状态信息通过第三接口反馈给总控模块, 将机器人的状态信息通过第二接口反馈给算法模块; 其中, 所述电机的状态信息包括位置、速度和力矩等; 所述机械臂的状态信息包括关节角度、关节角速度、关节角加速度、末端位姿、末端线速度、末端角速度、末端线加速度和末端线加速度等。

[0202] 为了更加清晰本发明的实施例的技术方案, 下面阐述采用本发明的硬件和软件环境应用实例:

[0203] 参考图5所示, 图5是一应用实例的机器人控制系统的硬件结构模型, 在机器人控制器的搭建软件架构, 运行于Linux操作系统, 该Linux主机可以是X86架构的PC机, 或者ARM芯片嵌入式架构的开发板, 总控模块接入控制端的人机交互界面的控制指令。

[0204] Linux主机可以安装如下软件: 安装Xenomai或者RTAI或者RT Preempt的实时内核补丁; 安装ROS, OROCOS, rFSM等软件。

[0205] 参考图6所示, 图6是基于ROS和OROCOS的搭建的软件架构图; 在控制过程中, 操作系统上运行总控模块、算法模块和通信管理模块。

[0206] 1、对于总控模块:

[0207] (1) 总控模块使用ROS的orocreate-catkin-pkg方法创建ROS的Package, 记为Ec_control_system, 然后在Package中, 通过继承OROCOS的RTT::TaskContext类, 记为Ec_control_system_component。

[0208] 在Ec_control_system_component类的构造函数中, 设置为执行如下操作:

[0209] a) 利用OROCOS的RTT::Input与RTT::Output方法, 对模块的输入, 输出接口进行定义。

- [0210] 其中输入的接口包括：
- [0211] ①通信管理模块传入的诊断数据；
- [0212] ②通信管理模块传入的状态反馈信息：包括电机运行状态等等；
- [0213] ③控制器状态机的状态；
- [0214] 输出的接口包括：
- [0215] ①控制器状态机事件触发，输出给控制器状态机。
- [0216] b) 利用OROCOS的Operational Caller方法设置函数调用接口。
- [0217] 第一，设置事件报告的回调函数：对事件报告处理请求进行响应，包括生成错误的时间戳，事件级别等信息，并将事件信息，发送给人机交互界面显示。
- [0218] 第二，设置警报设置的回调函数：依据诊断信息，判断是否生成警报。比如，位置，速度，加速度是否超限等等。
- [0219] 第三，设置各种运动规划的控制指令触发函数，这些函数将对算法模块的相应响应函数进行调用。
- [0220] c) 调用OROCOS的Properties方法定义总控模块的属性，将总控模块定义一个机械臂关节个数的属性。
- [0221] (2) 在Ec_control_system_component的StartHook() 成员函数中，设置执行如下操作：
- [0222] a) 检查日志报告是否正常，若异常直接退出，并将相关信息通过事件报告接口传递给总控模块处理；
- [0223] b) 通过ICE开发的通信协议建立与人机交互界面的通信连接，调用通信协议提供的动态异步远程过程调用(RPC)方法，对人机交互界面发起的控制指令进行响应的回调函数进行绑定。该回调函数，首先根据ICE(The Internet Communications Engine，互联网通信引擎)提供的远程过程异步调用方法传入的第一个参数，判断调用类型，然后依据此类型选择调用相应运动规划的运动指令触发函数。
- [0224] (3) 对于Ec_control_system_component的CleanUpHook() 成员函数，为了使得该函数在总控模块结束运行时，实现自动调用，还可以设置为执行如下操作：
- [0225] a) 调用ICE开发的通信协议接口，关闭与人机交互界面的通信连接。
- [0226] (4) 对于控制器状态机，参考图7所示，图7是控制器状态机的状态变化示意图；可以设置Init、Ready、Fault、Active.Recovery、Active.Halt、Active.Hands、Active.ToZero、Active.PTP、Active.Line、Active.Circle、Active.Stop共十一个状态，分别代表初始化、等待指令输入、恢复、暂停、手动示教、回到原点、暂停运动、暂停运动、暂停运动、急停状态。其中，Active.Recovery、Active.Halt、Active.Hands、Active.ToZero、Active.PTP、Active.Line、Active.Circle、Active.Stop这八个状态组成一个Active状态的集合，Active的状态转移规则，对八个子状态均有效。例如，给八个状态中的任一个，写入“e_ready”事件，将控制器状态机的状态从当前状态转移到Ready状态(即等待指令输入状态)。
- [0227] 另外，还可以使用Lua语言，编写总控模块的启动文件，该启动文件设置为执行如下动作：
- [0228] a) 通过OROCOS的import方法，加载模块进行运行；

- [0229] b) 定义模块的刷新频率,线程的优先级别;
- [0230] c) 对模块的属性进行赋值;
- [0231] d) 通过OROCOS的connect方法,将总控模块的输入、输出接口和算法模块和通信管理模块的接口建立连接。
- [0232] e) 通过OROCOS的start方法,运行总控模块,总控模块将先调用StartHook()函数,然后按预设刷新频率,实时地周期性调用UpdateHook()函数。
- [0233] 2、对于算法模块:
- [0234] 算法模块使用ROS的orocreate-catkin-pkg方法创建ROS的Package,记为Ec_control_loop,然后在Package中,通过继承OROCOS的RTT::TaskContext类,记为Ec_control_loop_component。
- [0235] (1) 在Ec_control_loop_component类的构造函数中,设置为执行如下操作:
- [0236] a) 算法模块利用OROCOS的RTT::Input与RTT::Output方法,对输入,输出接口进行定义。
- [0237] 其中输入的接口包括:
- [0238] ①通信管理模块传入的电机运行数据;
- [0239] ②通信管理模块传入的诊断数据;
- [0240] ③设备状态机的状态;
- [0241] ④控制器状态机的状态;
- [0242] 输出的接口包括:
- [0243] ①电机控制指令数据,输出给设备通信模块;
- [0244] ②设备状态机事件触发,输出给设备状态机;
- [0245] ③控制器状态机事件触发,输出给控制器状态机。
- [0246] b) 利用OROCOS的Operational Caller方法设置函数调用接口,设置事件报告的接口:该接口将触发总控模块的事件报告处理函数设置各种运动规划的返回原点指令响应函数。
- [0247] c) 调用OROCOS的Properties方法定义算法模块的属性,算法模块定义一个机械臂关节个数的属性。
- [0248] (2) 在Ec_control_loop_component的StartHook()成员函数中,设置为执行如下操作:
- [0249] a) 检查日志报告是否正常,若异常直接退出,并将相关信息通过事件报告接口传递给总控模块处理;
- [0250] b) 检查电机运行数据通道是否有数据,若无数据直接退出,并将相关信息通过事件报告接口传递给总控模块处理。
- [0251] (3) 对于Ec_control_loop_component类的UpdateHook()成员函数,设置该函数在算法模块运行时,按照用户设定的频率实时运行(如设为100Hz),可以设置为执行如下操作:
- [0252] a) 读取控制器状态机状态;
- [0253] b) 根据控制器状态机的不同状态,执行不同操作:
- [0254] I、如果是暂停运动,暂停运动,暂停运动,手动示教,急停,回到原点状态。此时,执

行如下操作：

[0255] 如果指令缓冲区的控制指令个数小于20个，则将所有指令一起发送给通信管理模块，并将控制器状态机的状态改变为等待指令输入状态；

[0256] 如果指令缓冲区的控制指令个数大于20个，则取指令队列末尾的20个，发送给通信管理模块；

[0257] II、如果是暂停状态，则什么也不做。

[0258] (4) 对于Ec_control_loop_component类，定义暂停运动，暂停运动，暂停运动，手动示教，急停，回到原点，暂停，恢复等函数调用接口，实现如下：

[0259] a) 暂停运动，暂停运动，暂停运动，手动示教，回到原点的函数，内部实现如下：

[0260] 检查控制器状态机是否处于等待指令输入状态。如果不是则退出，并将相关信息通过事件报告接口传递给总控模块处理；

[0261] 读取电机的当前的状态信息；

[0262] 依据电机的当前状态，分别调用暂停运动，暂停运动，暂停运动，手动示教，回到原点的运动规划，并将生成的电机控制指令保存到指令缓冲区；

[0263] 将控制器状态机设为相应状态。比如暂停运动回调函数，则将控制器状态机设为暂停运动状态。

[0264] b) 暂停函数，内部实现如下：

[0265] 检查控制器状态机是否是暂停运动，暂停运动，暂停运动，手动示教，回到原点等状态。如果不是则退出，并将相关信息通过事件报告接口传递给总控模块进行处理；

[0266] 记录当前控制器状态机的当前状态，并将控制器状态机的状态转变为暂停状态。

[0267] c) 恢复函数，内部实现如下：

[0268] 检查控制器状态机是否是暂停状态。如果不是则退出，并将相关信息通过事件报告接口传递给总控模块处理；

[0269] 将控制器状态机的状态转变为暂停前的状态。

[0270] d) 急停函数，内部实现如下：

[0271] I、检查控制器状态机是否是暂停运动，暂停运动，暂停运动，手动示教，回到原点等状态。如果不是则退出，并将相关信息通过事件报告接口传递给总控模块处理；

[0272] II、读取电机的当前的状态信息；

[0273] III、将电机控制指令缓冲区清零；

[0274] IV、调用速度规划运动规划，让电机以最短时间，速度降为0，并将生成的电机控制指令保存到指令缓冲区。

[0275] (5) 使用Lua语言，编写算法模块的启动文件，设置为执行如下动作：

[0276] a) 通过OROCOS的import方法，加载算法模块；

[0277] d) 定义算法模块的刷新频率，线程的优先级别；

[0278] e) 对算法模块的属性进行赋值；

[0279] d) 通过OROCOS的connect方法，将算法模块的输入，输出接口和总控模块和通信管理模块的接口建立连接。

[0280] e) 通过OROCOS的start方法，运行算法模块，算法模块先调用StartHook()函数，然后按设置的刷新频率，实时地周期性调用UpdateHook()函数。

[0281] 3、对于通信管理模块：

[0282] 通信管理模块可以通过linux主机minicom中的ttyACM0与Arm开发板通信，可以在该Arm开发板上运行一个CANOpen主站协议，该主站协议可以设置一个指令缓存区，最多可以存储25个指令。

[0283] 通信管理模块可以利用OROCOS的RTT::Input与RTT::Output方法与机器人算法模块和总控模块进行通信。

[0284] 利用rFSM软件建立设备状态机，对通信管理模块的业务逻辑进行控制。

[0285] 通信管理模块利用OROCOS的RTT::Input与RTT::Output方法与设备状态机连接，可改变设备状态机的状态，并读取状态。

[0286] (1) 通信模块使用ROS的orocreate-catkin-pkg方法创建为ROS的Package，然后在Package中，通过继承OROCOS的RTT::TaskContext类，创建一个OROCOS的实时模块，记为Ec_component。

[0287] 在Ec_component类的构造函数中，设置为执行如下操作：

[0288] a) 通信管理模块利用OROCOS的RTT::Input与RTT::Output方法，对输入，输出接口进行定义。

[0289] 其中输入的接口包括：

[0290] ①算法模块传入的控制指令数据；

[0291] ②设备状态机的状态；

[0292] 输出的接口包括：

[0293] ①诊断数据，输出至算法模块和总控模块；

[0294] ②电机运行数据和机械臂状态数据，输出给算法模块；

[0295] ③状态机事件触发，输出给设备状态机。

[0296] b) 利用OROCOS的Operational Caller方法定义函数调用接口，通信管理模块定义事件报告的接口，通过该接口触发总控模块的事件报告处理函数。

[0297] c) 调用OROCOS的Properties方法定义通信管理模块的属性，通信管理模块定义一个机械臂关节个数的属性。

[0298] (2) 在Ec_component的StartHook()成员函数中，设置为执行如下操作：

[0299] a) 检查日志报告是否正常，若异常直接退出，并将相关信息通过事件报告接口传递给总控模块进行处理；

[0300] b) 电机驱动初始化：

[0301] I、通过ttyACM0与电机驱动器建立通信；

[0302] II、电机指令缓冲队列清空；

[0303] III、电机使能，如果使能成功，则进行下一步，否则退出；

[0304] IV、读取电机的位置，计算机器人的机械臂当前状态，包括关节角度，机械臂末端位姿；

[0305] c) 机械臂状态初始化：

[0306] 依据电机位置，判断机械臂是否需要执行回零运动。如果机械臂任一关节角度与零度相差大于0.01度，则执行回零运动，调用暂停运动规划，对回零运动进行规划。

[0307] d) 改变设备状态机的状态：

- [0308] 如果机器人的机械臂需要执行回零运动,则控制设备状态机保持Init状态不变;否则,向设备状态机发送“e_nominal”事件,将设备状态机的状态转换为Active.Nominal。
- [0309] (3) 对于Ec_component的UpdateHook()成员函数,设置该函数在通信管理模块运行时,以用户设定的频率实时运行(如设为1KHz),设置为执行如下操作:
- [0310] a) 读取设备状态机状态;
- [0311] b) 根据设备状态机的不同状态,执行不同操作:
- [0312] I、如果是Init状态,执行机械臂的回零运动。此时,执行如下操作:
- [0313] 读取系统的时钟,依据回零运动轨迹规划结果,计算该时刻电机的运动指令,并将该单条运动指令发送给CANOpen主站。
- [0314] 如果运动已回零成功,向设备状态机发送“e_nominal”事件,将设备状态机转换为Active.Nominal,并将该事件报告给总控模块。
- [0315] II、如果是Active.Nominal状态。此时,执行如下操作:
- [0316] 从控制指令数据的输入通道中,读取控制指令,并存储到电机指令缓冲队列中。
- [0317] 读取CANOpen主站指令缓冲区的现有指令个数,如果小于10个,则一次性从电机指令缓冲队列中取出15个运动指令发送给CANOpen主站。如果电机指令缓冲队列的指令个数小于15个,则一次性全部发送给CANOpen主站。
- [0318] III、如果是Active.Recovery状态。此时,通信管理模块处于恢复状态。
- [0319] 此时,依据诊断信息,对系统状态进行恢复,若恢复成功,向设备状态机发送“e_nominal”事件,将设备状态机转换为Active.Nominal。并将该事件报告给总控模块。
- [0320] 若恢复不成功,给状态机发送“e_fault”事件,状态机转换为Fault状态,将该事件报告给总控模块,并直接退出UpdateHook()。
- [0321] IV、如果是Active.Halt状态。此时,模块处于暂停状态,执行如下操作:检查控制指令数据的输入通道中是否有新的指令,若有新指令,则读取控制指令,并存储到电机指令缓冲队列中。
- [0322] V、如果是Active.Hands状态。此时,模块处于手控模式,执行如下操作:
- [0323] 读取系统的时钟,依据运动轨迹规划结果,计算该时刻电机的运动指令,并将该单条运动指令发送给CANOpen主站。
- [0324] VI、如果是Fault状态,则直接退出UpdateHook()。
- [0325] c) 读取电机状态,根据机械臂的模型计算机械臂关节以及末端运动状态信息,并通过输出数据通道,传递给算法模块与总控模块;
- [0326] d) 检查控制主站是否有错误报告信息,如果有错误信息,则将诊断信息传递给算法模块与总控模块。向设备状态机发送“e_recovery”事件,将设备状态机转换为Active.Recovery状态、并将该事件报告给总控模块。
- [0327] (4) 对于Ec_component的CleanUpHook()成员函数,该函数在模块结束运行时,自动调用,设置执行如下操作:
- [0328] a) 关闭电机驱动使能;
- [0329] b) 关闭电机驱动连接。
- [0330] (5) 参考图8所示,图8是设备状态机的状态变化示意图。共有Init(初始化)、Fault(中断)、Active.Recovery(恢复)、Active.Hands(手动示教)、Active.Halt(暂停)、

Active.Nominal(运转)六个状态。Active.Recovery、Active.Hands、Active.Halt、Active.Nominal四个状态组成一个Active(使能)的状态集合,Active的状态转移规则,对四个子状态均有效。

- [0331] (6) 使用Lua语言,编写该模块的启动文件,该启动文件设置执行如下动作:
 - [0332] a) 通过OROCOS的import方法,加载通信管理模块;
 - [0333] b) 定义通信管理模块的刷新频率,线程的优先级别;
 - [0334] c) 对通信管理模块的属性进行赋值;
 - [0335] d) 通过OROCOS的connect方法,将通信管理模块的输入,输出接口和总控模块和算法模等的接口建立连接。
 - [0336] e) 通过OROCOS的start方法,运行通信管理模块,通信管理模块先调用StartHook()函数,然后按定义好的刷新频率,实时地周期性调用UpdateHook()函数。
- [0337] 对于上述总控模块、算法模块和通信管理模块,设置为当运行后,如果用户需要中途停止该模块,同时按键盘的ctrl键与D键。
- [0338] 综上实施例,基于ROS和OROCOS,利用OROCOS的实时性特点,保证软件程序的实时性能;充分利用ROS的开放性,基于ROS、OROCOS开发的总控模块、算法模块和通信管理模块进行实时通信,共同构成一个完整的机器人控制器软件;进一步建立了控制器状态机和设备状态机,从而实现对总控模块、通信管理模块的业务逻辑进行有效管理。
- [0339] 通过OROCOS的RTT::Input,RTT::Output方法建立总控模块、算法模块和通信管理模块的数据输入、输出通道,通过OROCOS的Operational Caller方法定义函数调用接口,通过OROCOS的Properties方法定义总控模块、算法模块和通信管理模块的属性。因此,保证了总控模块、算法模块和通信管理模块之间的独立性与解耦性。
- [0340] 以上所述实施例的各技术特征可以进行任意的组合,为使描述简洁,未对上述实施例中的各个技术特征所有可能的组合都进行描述,然而,只要这些技术特征的组合不存在矛盾,都应当认为是本说明书记载的范围。
- [0341] 以上所述实施例仅表达了本发明的几种实施方式,其描述较为具体和详细,但并不能因此而理解为对发明专利范围的限制。应当指出的是,对于本领域的普通技术人员来说,在不脱离本发明构思的前提下,还可以做出若干变形和改进,这些都属于本发明的保护范围。因此,本发明专利的保护范围应以所附权利要求为准。

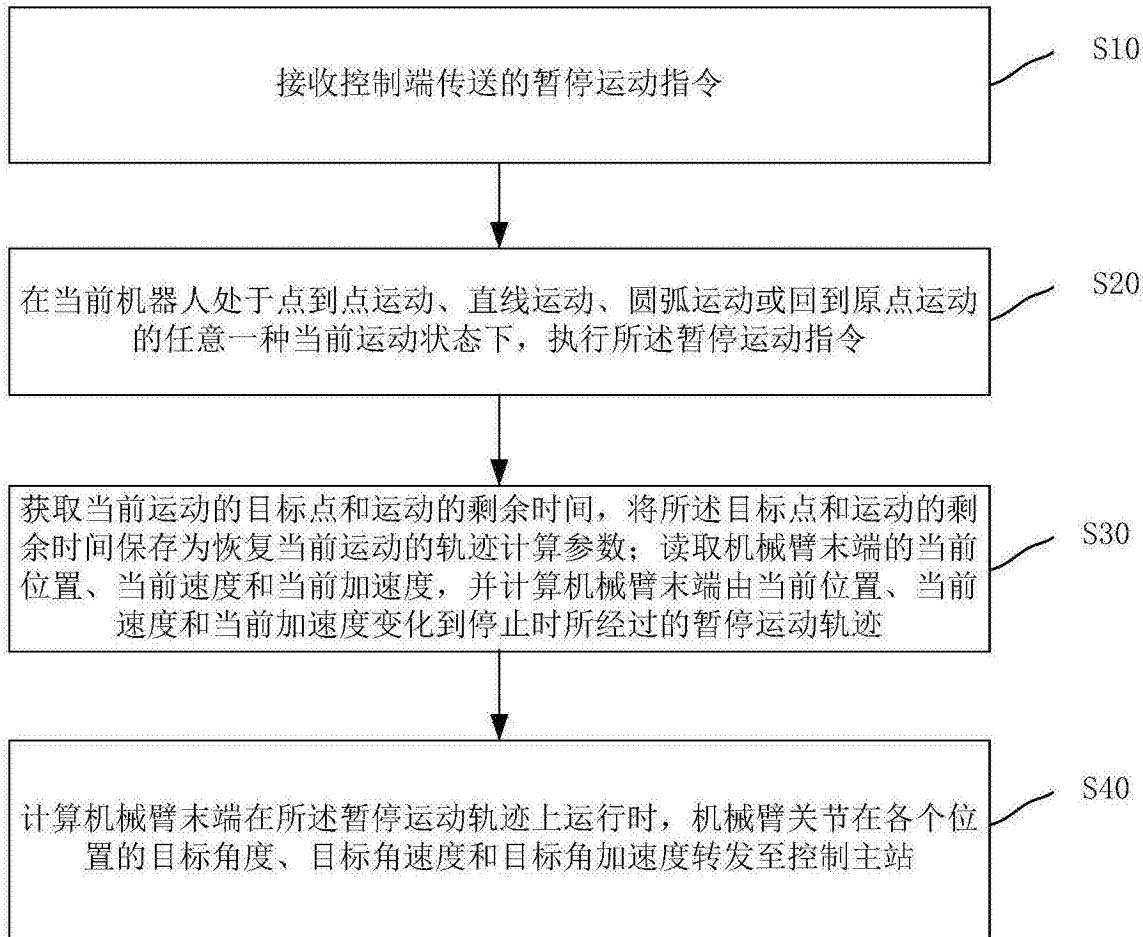


图1

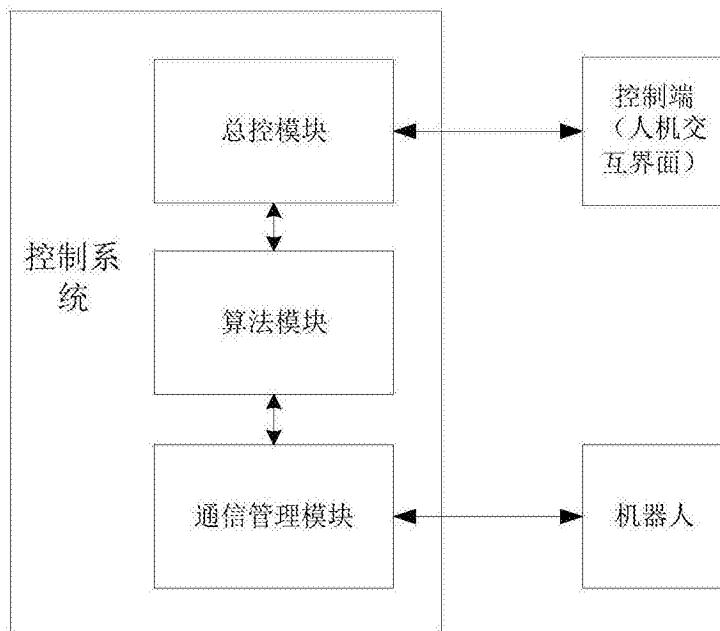


图2

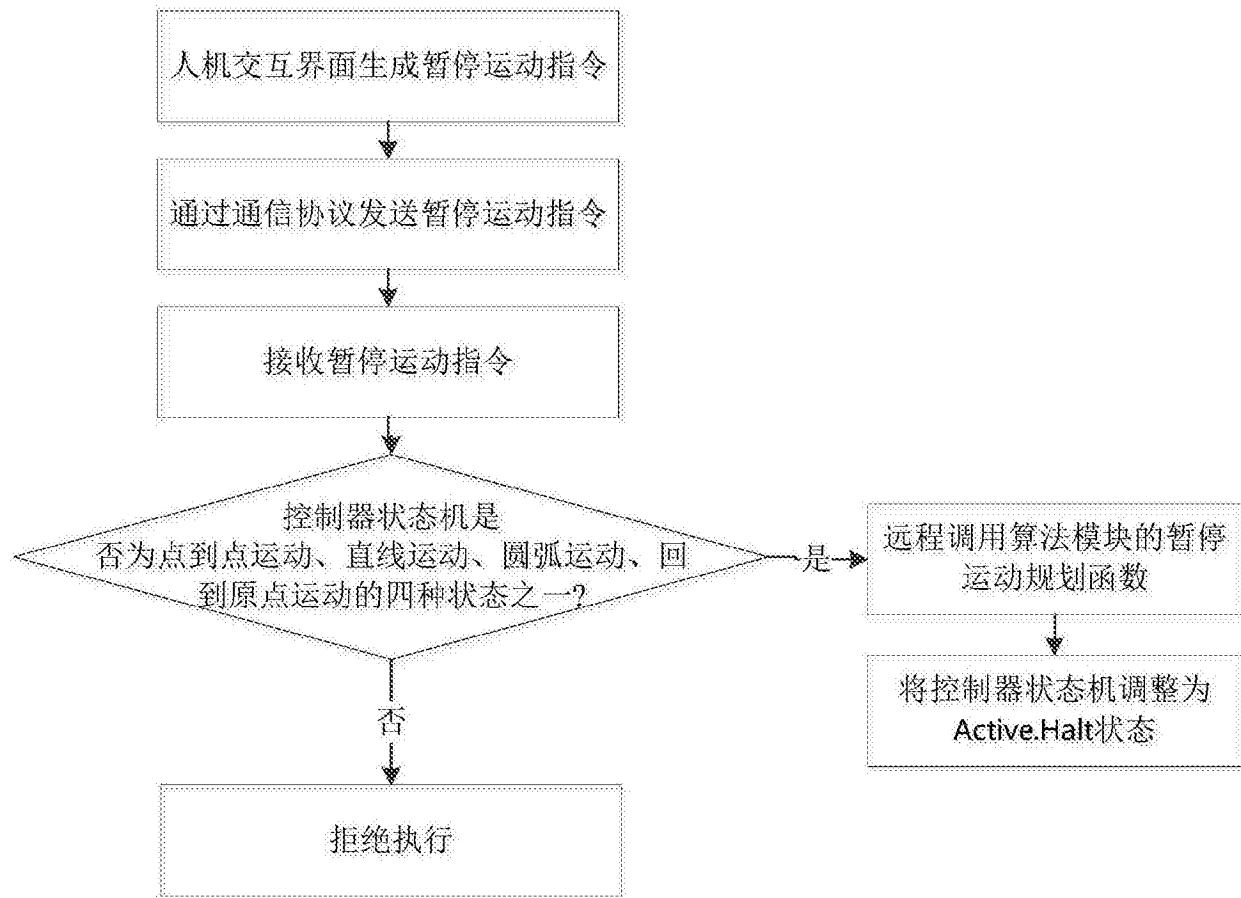


图3

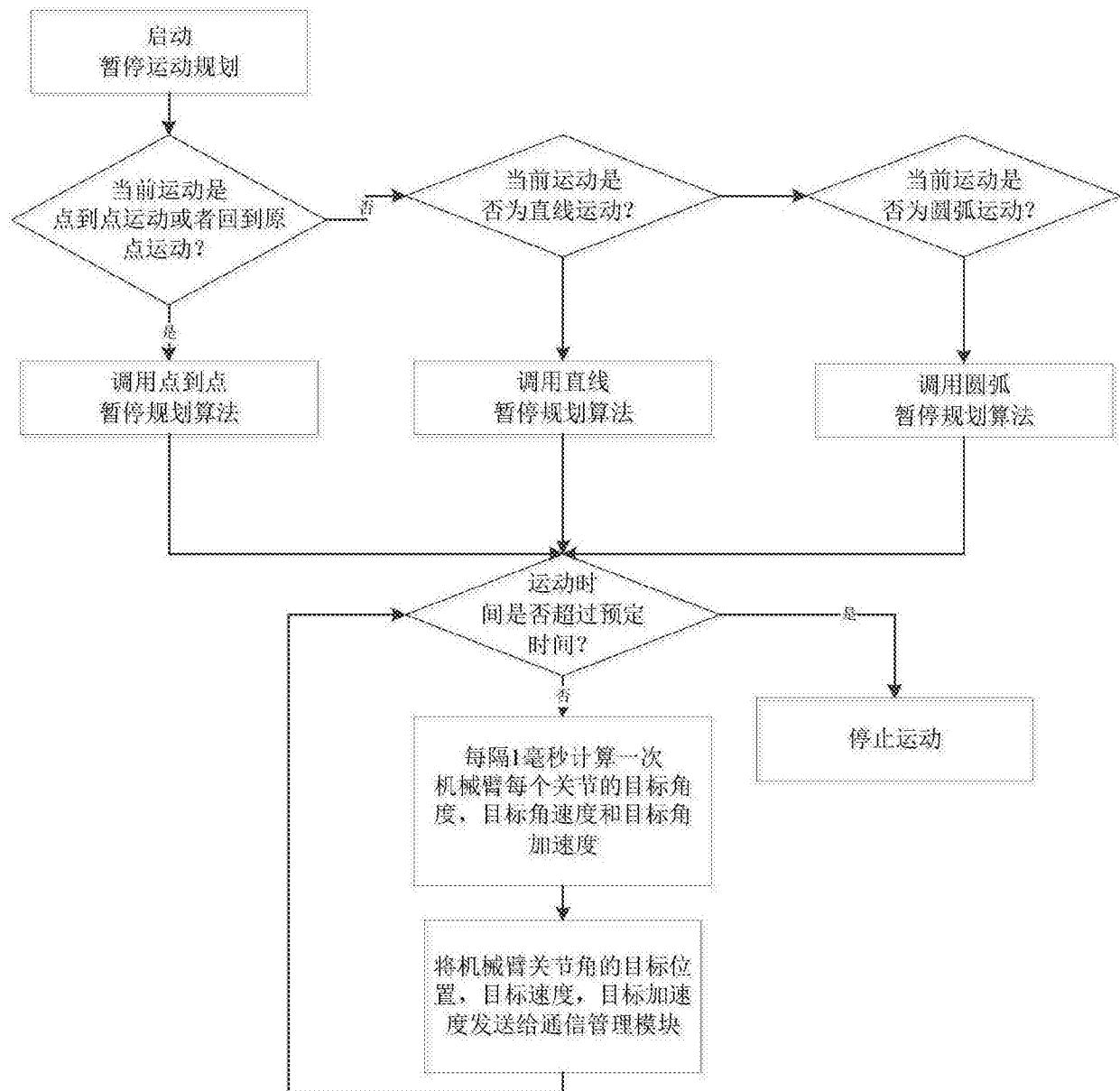


图4

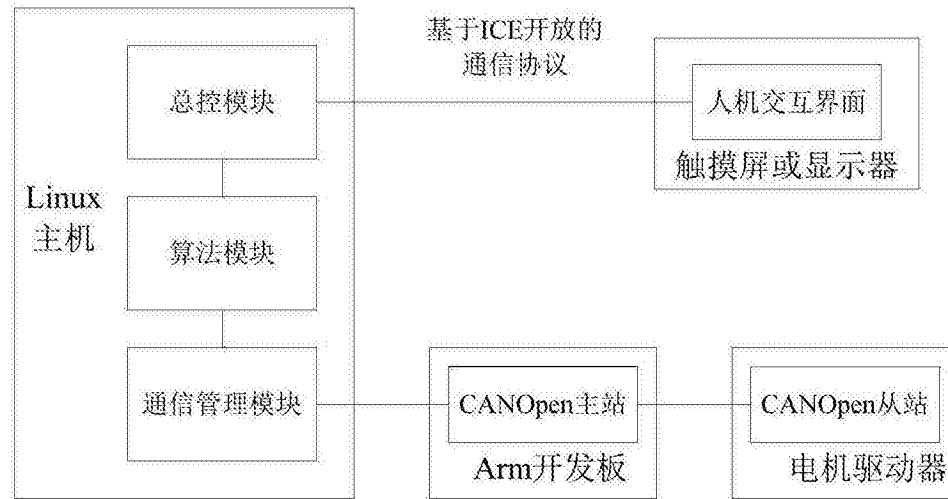


图5

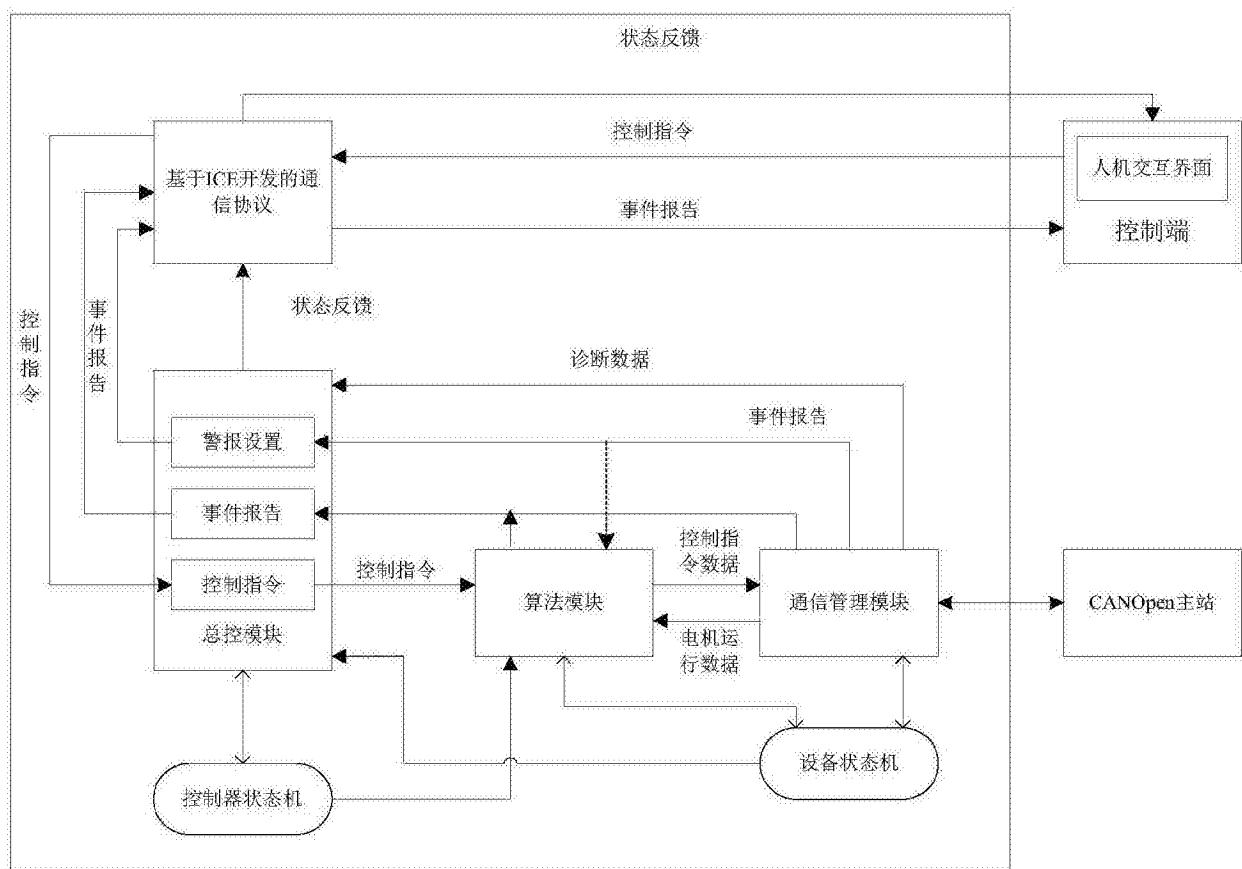


图6

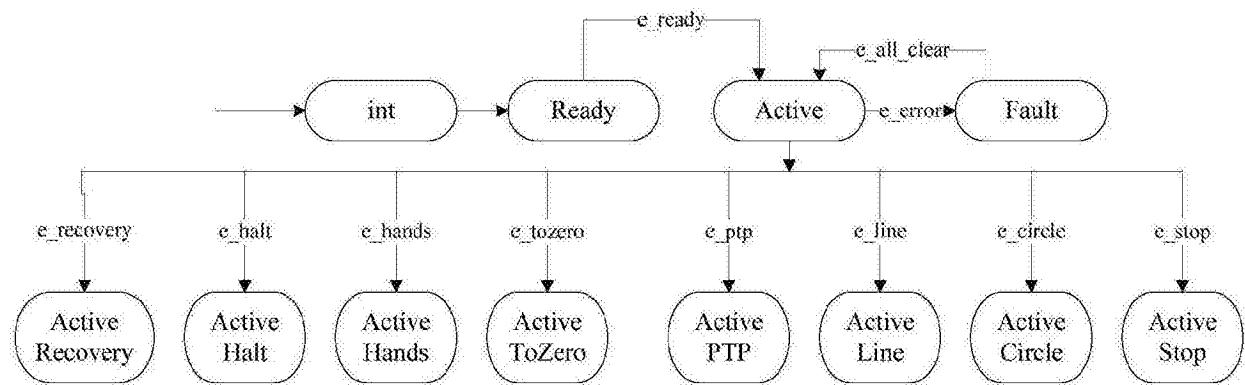


图7

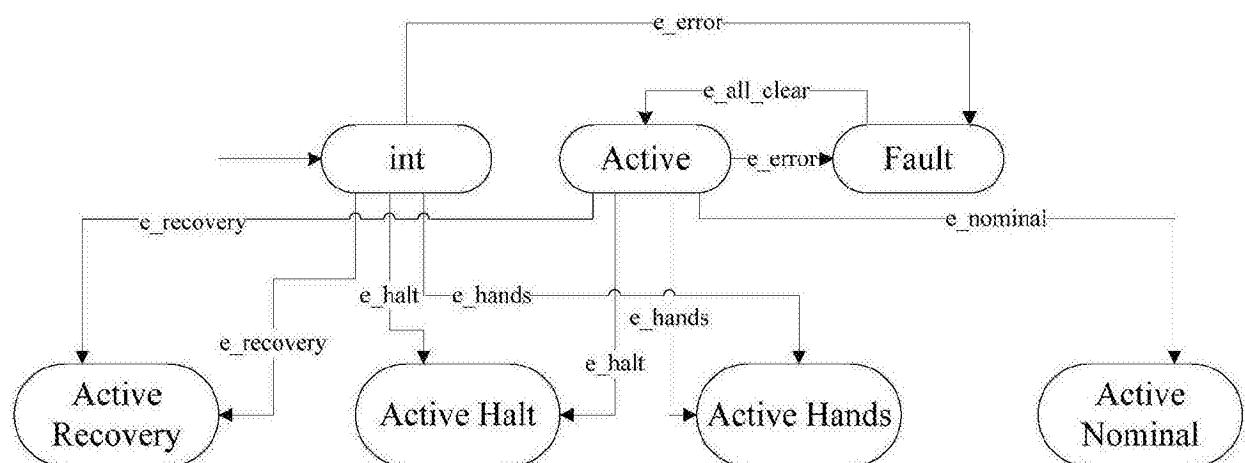


图8