



(12)发明专利

(10)授权公告号 CN 104786221 B

(45)授权公告日 2016.12.07

(21)申请号 201510172297.4

(51)Int.Cl.

(22)申请日 2015.04.13

B25J 9/16(2006.01)

(65)同一申请的已公布的文献号

审查员 李方芬

申请公布号 CN 104786221 A

(43)申请公布日 2015.07.22

(73)专利权人 浙江工业大学

地址 310014 浙江省杭州市下城区朝晖六区潮王路18号

专利权人 浙江金火机床有限公司

(72)发明人 董辉 金阔洋 赖宏焕 韩瑞祥

楼杰

(74)专利代理机构 杭州斯可睿专利事务所有限

公司 33241

代理人 王利强

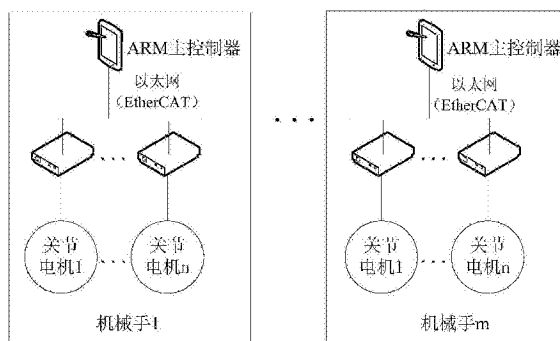
权利要求书2页 说明书6页 附图3页

(54)发明名称

一种基于以太网的开放式机械手控制方法

(57)摘要

一种基于以太网的开放式机械手控制方法,包括如下步骤:1)采用基于ARM Cortex-A8内核的嵌入式工控板并搭载有CoDeSysRTS软件环境的嵌入式Linux作为整个技术方案的实现平台;2)通过EtherCAT协议的以太网连接ARM控制器与各个关节电机的伺服驱动器;3)采用以太网与PC端的网卡相连,传输并在PC端显示当前机械手的运行状态;4)通过控制器的开放式直接编程实现对机械手的笛卡尔坐标的轨迹规划与速度控制;采用机器人逆运动学解算法将笛卡尔坐标转换为各个关节电机的转速转加速度,通过以太网读取机械手各个关节状态变量并通过实时的速度规划计算控制信息从而进行位置误差补偿。本发明实时性良好、稳定性较高、开放性良好。



1. 一种基于以太网的开放式机械手控制方法,其特征在于:包括如下步骤:

1)采用基于ARM Cortex-A8内核的嵌入式工控板并搭载有CoDeSysRTS软件环境的嵌入式Linux作为整个技术方案的实现平台;

2)通过EtherCAT协议的以太网连接ARM控制器与各个关节电机的伺服驱动器,实现对机械手各个关节的状态读取与控制;

系统已将EtherCAT的电机驱动标准的数据包封装成系统变量,通过修改MC_MOVEABSOLUTE模块命令电机到一个指定的绝对位置;所述的MC_MOVEABSOLUTE模块包含可以设置命令执行项Execute的输入端、指定电机运动的目标位置Position、设定电机的速度Velocity、电机的加速度Acceleration及电机的减速度Deceleration和电机转向Direction,所述模块输出端查看电机的工作状态,通过修改MC_MOVEVELOCITY模块使电机以一定的速度转动,其输入端同时也具有执行、速度、加减速设置项,输出端查看各种电机运行状态的参数;通过修改MC_POWER可以控制电机的启停,输入端有使能、电源平台开启项及驱动开始项,输出项也是电机各种预开启状态的参数;

模块MC_READACTUALPOSITION来返回电机转动的实际位置,输入为使能控制,输出返回电机的实际状态或电机的出错状态信息;

3)采用以太网与PC端的网卡相连,传输并在PC端显示当前机械手的运行状态;

4)通过控制器的开放式直接编程实现对机械手的笛卡尔坐标的轨迹规划与速度控制;

采用机器人逆运动学解算法将笛卡尔坐标转换为各个关节电机的转速和加速度,通过以太网读取机械手各个关节状态变量并通过实时的速度规划计算控制信息从而进行位置误差补偿。

2. 如权利要求1所述的一种基于以太网的开放式机械手控制方法,其特征在于:在步骤3)中,所述的以太网与PC端网卡相连显示当前机械手运行状态,所述的以太网传输的数据报文包括机械手的所处的工作流程的具体步骤、工作状态、执行器的笛卡尔坐标及其姿态向量,所述数据报文格式包含数据帧头、数据长度、机械手的ID、功能号、数据流以及校验位,PC端监控器能够增删连接的机械手设备,并实时显示各个机械手状态,随时对某个机械手进行暂停和复位操作。

3. 如权利要求1所述的一种基于以太网的开放式机械手控制方法,其特征在于,在步骤4)中,所述的开放式直接编程包括如下实现途径:a、用户能够在人机界面中直接编写C语言或者机器语言代码从而自主规划机械手的运动轨迹;b、向控制器中导入CAD轨迹图直接生成能够描述相应空间轨迹的机器语言。

4. 如权利要求1所述的一种基于以太网的开放式机械手控制方法,其特征在于,在步骤4)中,轨迹规划和速度控制过程如下:

在X/Y坐标上存在 p_{spe} 的轨迹,它与X的横坐标存在 θ 的夹角,对该轨迹进行时间分割插补后得到三个插补点 p_1 、 p_2 、 p_3 ,机械手从轨迹的第一个插补点 $p_1(x_1, y_1)$ 以速度 $v_1(v_{1x}, v_{1y})$ 按照预先设定的轨迹运动到第二个插补点 $p_2(x_2, y_2)$,在一个插补周期后,即在 $p_2(x_2, y_2)$ 点处进行实时位置反馈,此时发现由于Y轴负载的变化导致Y轴的速度比实际规划的速度要小,所以在一个插补周期后导致机械手未到指定的点 p_2 上,而是在点 p_2^* 上,最后产生的位置误差是 $p_\Delta(x_\Delta, y_\Delta)$,

$$\begin{aligned} x_{\Delta} &= x_2 - x_2^{\bullet} \\ y_{\Delta} &= y_2 - y_2^{\bullet} \end{aligned} \quad (1)$$

同时通过计算得到实际的X轴和Y轴的速度 $v_1^{\bullet}(v_{1x}^{\bullet}, v_{1y}^{\bullet})$,对比实际的速度和规划的速度比较各自轴的速度快慢,假设实际速度 v_{1y}^{\bullet} 与规划速度 v_{1y} 有较大的差距,则将Y轴作为下次速度规划参考轴,这时的点 $p_2(x_2, y_2)$ 的速度 $v_2(v_{2x}, v_{2y})$:

$$\begin{cases} v_{2x} = (v_{2y} * T - x_{\Delta}) / (T * \tan \theta) \\ v_{2y} = v_{1y}^{\bullet} \end{cases} \quad (2)$$

从而得出插补点 p_2 的速度,结合插补周期 T ,得到新的插补点 p_m 。

一种基于以太网的开放式机械手控制方法

技术领域

[0001] 本发明应用于机械手运动控制领域,涉及一种基于以太网的开放式机械手控制方法。

背景技术

[0002] 在科技日新月异的今天,机械手的应用领域越来越广泛。它可以重复不断的劳动和工作,并且不知疲劳、不受工作环境的限制,能够胜任危险环境的工作,提高劳动生产效率,降低生产成本。它可以通过编程来完成各种预期的作业任务,在构造和性能上兼顾人和机器的优点,具有人的适应性和智能。机械手作业的准确性及其在各种环境下完成任务的能力是重点研究方向。

[0003] 目前的工业机器人主要在焊接、喷涂、搬运以及堆垛等重复性和劳动强度极大的行业有着广泛的应用。但是传统机械手一般使用脉冲方式控制关节电机,存在抗干扰能力差、无自主纠错能力、应用范围受限的不足,并且不具备开放式编程的能力,通用性不强以及控制关机数量有限的不足。在实际的应用中还存在连线过多导致的安装复杂等不足。

发明内容

[0004] 为了克服目前机械手控制存在的实时性较差、稳定性较低以及开放度较差的不足,以及针对高要求、高速度、高精度的运动控制的需要,本发明提供了一种实时性良好、稳定性较高、开放性良好的基于以太网的开放式机械手控制方法。

[0005] 本发明解决其技术问题所采用的技术方案是:

[0006] 一种基于以太网的开放式机械手控制方法,包括如下步骤:

[0007] 1)采用基于ARM Cortex-A8内核的嵌入式工控板并搭载有CoDeSysRTS软件环境的嵌入式Linux作为整个技术方案的实现平台;

[0008] 2)通过EtherCAT协议的以太网连接ARM控制器与各个关节电机的伺服驱动器,实现对机械手各个关节的状态读取与控制;

[0009] 3)采用以太网与PC端的网卡相连,传输并在PC端显示当前机械手的运行状态;

[0010] 4)通过控制器的开放式直接编程实现对机械手的笛卡尔坐标的轨迹规划与速度控制;

[0011] 采用机器人逆运动学解算法将笛卡尔坐标转换为各个关节电机的转速转加速度,通过以太网读取机械手各个关节状态变量并通过实时的速度规划计算控制信息从而进行位置误差补偿。

[0012] 进一步,在步骤1)中,采用基于ARM Cortex-A8内核的嵌入式工控板并搭载有CoDeSysRTS实时运系统的嵌入式Linux作为整个技术方案的平台。

[0013] 再进一步,在步骤3)中,系统已将EtherCAT的电机驱动标准的数据包封装成系统变量,通过修改MC_MOVEABSOLUTE模块命令电机到一个指定的绝对位置;所述的MC_MOVEABSOLUTE模块包含可以设置命令执行项Execute的输入端、指定电机运动的目标位置

Position、设定电机的速度Velocity、电机的加速度Acceleration及电机的减速度Deceleration和电机转向Direction,所述模块输出端查看电机的工作状态,通过修改MC_MOVEVELOCITY模块使电机以一定的速度转动,其输入端同时也具有执行、速度、加减速设置项,输出端查看各种电机运行状态的参数;通过修改MC_POWER可以控制电机的启停,输入端有使能、电源平台开启项及驱动开始项,输出项也是电机各种预开启状态的参数;

[0014] 模块MC_READACT-UALPOSITION来返回电机转动的实际位置,输入为使能控制,输出返回电机的实际状态或电机的出错状态信息。

[0015] 在步骤4)中,所述的以太网与PC端网卡相连显示当前机械手运行状态,所述的以太网传输的数据报文包括机械手的所处的工作流程的具体步骤、工作状态、执行器的笛卡尔坐标及其姿态向量。所述数据报文格式包含数据帧头、数据长度、机械手的ID、功能号、数据流以及校验位,PC端监控器能够增删连接的机械手设备,并实时显示各个机械手状态,随时对某个机械手进行暂停、复位等操作。

[0016] 在步骤5)中,所述的开放式直接编程包括如下实现途径:a、用户能够在人机界面中直接编写C语言或者机器语言代码从而自主规划机械手的运动轨迹;b、向控制器中导入CAD轨迹图直接生成能够描述相应空间轨迹的机器语言。

[0017] 在步骤5)中,轨迹规划和速度控制过程如下:在X/Y坐标上存在 p_{spe} 的轨迹,它与X的横坐标存在 θ 的夹角,对该轨迹进行时间分割插补后得到三个插补点 p_1 、 p_2 、 p_3 ,机械手从轨迹的第一个插补点 $p_1(x_1, y_1)$ 以速度 $v_1(v_{1x}, v_{1y})$ 按照预先设定的轨迹运动到第二个插补点 $p_2(x_2, y_2)$,在一个插补周期后,即在 $p_2(x_2, y_2)$ 点处进行实时位置反馈,此时发现由于Y轴负载的变化导致Y轴的速度比实际规划的速度要小,所以在一个插补周期后导致机械手未到指定的点 p_2 上,而是在点 p_2^* 上,最后产生的位置误差是 $p_\Delta(x_\Delta, y_\Delta)$,

$$\begin{aligned} x_\Delta &= x_2 - x_2^* \\ y_\Delta &= y_2 - y_2^* \end{aligned} \quad (1)$$

[0019] 同时通过计算得到实际的X轴和Y轴的速度 $V_1^*(v_{1x}^*, v_{1y}^*)$,对比实际的速度和规划的速度比较各自轴的速度快慢,假设实际速度 V_{1y}^* 与规划速度 V_{1y} 有较大的差距,则将Y轴作为下次速度规划参考轴,这时的点 $p_2(x_2, y_2)$ 的速度 $v_2(v_{2x}, v_{2y})$:

$$\begin{cases} v_{2x} = (v_{2y} * T - x_\Delta) / (T * \tan \theta) \\ v_{2y} = v_{1y}^* \end{cases} \quad (2)$$

[0021] 从而得出插补点 p_2 的速度,结合插补周期T,得到新的插补点 p_m 。

[0022] 本发明的有益效果主要表现在:采用嵌入式ARM平台作为机械手的主控制器,满足机械手控制算法的复杂性,并且使得控制器体积小巧,便于安装。控制器安装嵌入式Linux的操作系统并运行基于CoDeSys的软PLC运行平台的作为本发明的运行环境可以保证运行过程中的稳定性和实时性。采用EtherCAT协议的以太网作为控制数据主通路,可以实时更新分布式I/O口或轴的状态,支持多种拓扑,精确同步伺服轴进行协调运动且刷新周期短、效率更高。PC端监控器能够对机械手设备进行添加和删除,并且实时显示机械手的状态,控制机械手的启停和复位等操作,便于同时工作的多个机械手的统一管理调度。采用的实时

速度规划可以针对插补位置点对实际位置进行反馈,如有位置误差,可以对下一个插补周期内的速度进行重新规划,在有限的插补周期内消除误差,可以在一定范围内保证机械手的运行精度和机械轴的同步。

附图说明

- [0023] 图1是机械手控制网络结构框图。
- [0024] 图2是机械手控制器软件系统框图。
- [0025] 图3是机械手控制器硬件结构框图。
- [0026] 图4是实时速度规划图。
- [0027] 图5是电机控制流程图。

具体实施方式

- [0028] 下面结合附图对本发明作进一步描述。
- [0029] 参照图1~图5,一种基于以太网的开放式机械手控制方法,包括如下步骤:
- [0030] 1)采用基于ARM Cortex-A8内核的嵌入式工控板作为机械手的控制器;
- [0031] 2)控制器中运行装载有软PLC运行环境的嵌入式Linux的操作系统作为技术方案的运行环境;
- [0032] 3)通过EtherCAT协议的以太网连接ARM控制器与各个关节电机的伺服驱动器,实现对机械手各个关节的状态读取与控制;
- [0033] 4)采用以太网与PC端的网卡相连,传输并在PC端显示当前机械手的运行状态;
- [0034] 5)通过控制器的开放式直接编程实现对机械手的笛卡尔坐标的轨迹规划与速度控制的功能;
- [0035] 采用机器人逆运动学解算法将笛卡尔坐标转换为各个关节电机的转速转加速度等状态量。为减少机械手末端的位姿误差,提高机械手精度,本发明通过本发明通过以太网读取机械手各个关节状态变量并通过实时的速度规划计算控制信息从而进行位置误差补偿。
- [0036] 在步骤1)中,所述的ARM Cortex-A8内核的嵌入式工控板采用如下的功能模块:使用TI(美国德州仪器公司)的AM3354芯片作为主控芯片,并扩展512Mb的DDR3以及1Gb的NandFlash作为系统的存储设备。使用TPS650250作为电源管理的模块提供3个LDO电源,其中两个为3.3V一个为1.8V。并扩展2路千兆以太网作为主控制器与伺服驱动器的数据通信接口以及PC端通信的接口。通过芯片自带的LCD控制器以及ADC模块连接带有触摸屏的LCD,作为主要的人机交互模块。并流出4路USB接口用于连接鼠标、键盘和U盘等设备以及4个物理按键保证人机交互的多样性。以及2路UART接口,一路485总线,与外接的手持器通信。和12路的GPIO、一个SD卡槽作为扩展。这种结构既能保证功能又具有很好的扩展与适应性。上述的各类接口都或层叠或并列的排布在控制器的一侧,保证所有连线的有序性。
- [0037] 在步骤2)中,所述的软PLC运行环境为嵌入式Linux中运行的工业控制软件CoDeSys的运行平台,该环境能够解释在PC端使用IEC61131语言编写的代码,访问各个外部设备。所述的CoDeSys是一种跨平台的PLC软件编程工具。
- [0038] 在步骤3)中,所述的以太网采用EtherCAT(Ether Control Automation

Technology)协议。系统已将EtherCAT的电机驱动标准的数据包封装成系统变量,本发明通过修改MC_MOVEABSOLUTE模块命令电机到一个指定的绝对位置。所述的MC_MOVEABSOLUTE模块包含可以设置命令执行项Execute的输入端、指定电机运动的目标位置Position、设定电机的速度Velocity、电机的加速度Acceleration及电机的减速度Deceleration和电机转向Direction,所述模块输出端可以查看电机的工作状态,如忙碌Busy、错误Error等。发明通过修改MC_MOVEVELOCITY模块使电机以一定的速度转动,其输入端同时也具有执行、速度、加减速等设置项,输出端可以查看各种电机运行状态的参数。通过修改MC_POWER可以控制电机的启停,输入端有使能、电源平台开启项及驱动开始项(可以用来关闭quickstop装置),输出项也是电机各种预开启状态的参数。模块MC_READACTUALPOSITION来返回电机转动的实际位置,输入为使能控制,输出可以返回电机的实际状态或电机的出错状态等信息。还有与其他特殊功能的电机控制模块综合使用可以实现电机的各种复杂控制与监测。

[0039] 在步骤4)中,所述与PC端相连的以太网传输的数据报文包括机械手的所处的工作流程的具体步骤、工作状态、执行器的笛卡尔坐标及其姿态向量。所述数据报文格式包含数据帧头、数据长度、机械手的ID、功能号、数据流以及校验位。PC端监控器能够增删连接的机械手设备,并实时显示各个机械手状态。同时可以随时对某个机械手进行暂停、复位等操作。

[0040] 在步骤5)中,所述的开放式直接编程包括如下实现途径。a、用户能够在人机界面中直接编写C语言或者机器语言代码从而自主规划机械手的运动轨迹;b、向控制器中导入CAD轨迹图直接生成能够描述相应空间轨迹的机器语言。

[0041] 在步骤5)中,轨迹规划和速度控制的具体过程如下:

[0042] 在X/Y坐标上存在 p_{spe} 的轨迹,它与X的横坐标存在 θ 的夹角,对该轨迹进行时间分割插补后可以得到三个插补点 p_1 、 p_2 、 p_3 。机械手从轨迹的第一个插补点 $p_1(x_1, y_1)$ 以速度 v_1 (v_{1x}, v_{1y})按照预先设定的轨迹运动到第二个插补点 $p_2(x_2, y_2)$,在一个插补周期后,即在 $p_2(x_2, y_2)$ 点处进行实时位置反馈。此时可以发现由于Y轴负载的变化导致Y轴的速度比实际规划的速度要小,所以在一个插补周期后导致机械手未到指定的点 p_2 上,而是在点 p_2' 上。最后产生的位置误差是 $p_\Delta(x_\Delta, y_\Delta)$,

$$\begin{aligned} x_\Delta &= x_2 - x_2^* \\ y_\Delta &= y_2 - y_2^* \end{aligned} \quad (1)$$

[0044] 同时可以通过计算得到实际的X轴和Y轴的速度 $V_1^*(v_{1x}^*, v_{1y}^*)$,对比实际的速度和规划的速度可以比较各自轴的速度快慢。假设图中所示的实际速度 V_1^* 与规划速度 V_{1y} 有较大的差距,则将Y轴作为下次速度规划参考轴,这时的点 $p_2(x_2, y_2)$ 的速度 $v_2(v_{2x}, v_{2y})$:

$$\begin{cases} v_{2x} = (v_{2y} * T - x_\Delta) / (T * \tan \theta) \\ v_{2y} = v_{1y}^* \end{cases} \quad (2)$$

[0046] 从而得出插补点 p_2 的速度,结合插补周期T,可以得到新的插补点 p_m ,这在理论上消除了由于负载变化导致的轨迹误差,增加了机械手轨迹控制的准确性。当然在实际操作过程中,轨迹误差可以在一定的阈值范围存在,满足加工时所需的精度即可。

[0047] 参照图1,该图为本发明中基于以太网的机械手控制装置与方法的网络结构图。

[0048] 该嵌入式工控板采用ARM Cortex-A8的内核,使用TI(美国德州仪器公司)的AM3354芯片作为主控芯片,并集成了512MB的DDR3内存颗粒和1Gb的SLC Nandflash内存芯片。使用TPS650250的电源管理模块提供3个LDO电源,并扩展2路千兆以太网作为主控制器与伺服驱动器的数据通信接口以及PC端通信的接口。使用LCD控制器和ADC模块与带有触摸屏的LCD进行连接,提供人机交互操作界面。使用2路UART接口和一路485总线与外界手持器进行通信。流出的USB接口可以为鼠标、键盘和U盘等外部设备进行数据交换和操作,与4个物理按键保证了人机交互方式的多样性。同时提供了12路的GPIO、一个SD卡槽作为扩展以满足用户多样性的操作需求。

[0049] 参照图2,该图为机械手控制器软件系统框图。

[0050] 在嵌入式工控板中烧入Linux系统作为整个发明的运行环境,然后向其移入CoDeSysControl的Linux压缩包,使其在Linux上搭建好CoDeSysRTS实时运系统,用来编写机械手的可视化界面和算法的实现程序。通过人机界面可以方便使用者与机械手的交互,减少控制机械手的难度。在PC的CoDeSys平台下使用CFC语言搭建电机、速度控制、位置读取、角度控制等模块,实现机械手控制功能块平台。并且在CoDeSys设备根目录中配置EtherCAT主从站,建立好电机与CoDeSys软PLC平台之间的通信与控制。这样可以在该软PLC运行环境下,只需设置好相应主从站的网络配置和映射配置就可以实现控制伺服电机的转速、转向等。然后在Linux的CoDeSysRTS实时运系统环境下通过IEC61131-3语言实现速度实时规划算法,最终的返回结果会通过EtherCAT现场总线在机械手上得到响应,使机械手能根据预先设定的轨迹进行平滑的运动,实现了控制其运动状态的目的。通过CoDeSys软件界面对机械手进行配置时,在设备下选择“CoDeSys Control from zhengdo-ng”作为运行环境,并在其子树PLC下新建motor的主程序,然后将其拖入任务配置。上述操作结束后在新添设备中选择现场总线中的EtherCAT_M-master作为主站,并选择欧姆龙公司对应的伺服驱动器和相应的伺服电机。结束后,在主从站进行网络通信的配置。

[0051] 参照图3,机械手控制器硬件结构框图。

[0052] 整个硬件平台的搭建通过以太网使PC端监控器与ARM主控制器相连。以太网传输的数据报文包含机械手所处的工作流程的具体步骤、工作状态、执行器的笛卡尔坐标及其姿态向量。数据报文格式中包含数据帧头、数据长度、机械手ID、功能号、数据流以及校验位。ARM主控制器与伺服驱动器之间通过EtherCAT总线进行数据通讯,此EtherCAT的主站由植入嵌入式工控板的支持EtherCAT通讯的CoDeSys软件负责,负责网络管理与配置,起到系统控制作用。从站则为伺服驱动器、伺服电机及I/O通讯模块。在ARM主控制器上的CoDeSys通过向总线上的设备发送相应的指令,从站可以根据自己的地址取出相关的指令,根据设置的命令控制伺服电机的运动控制动作。同时通过PC端监控器能够增删连接的机械手设备,并实时显示各个机械手状态,可以随时对某个机械手进行暂停、复位等操作,保证整个运动过程的快速性和准确性。

[0053] 参照图4,该图为机械手实时速度规划原理图。

[0054] 在X/Y坐标上存在 $p_s p_e$ 的轨迹,它与X的横坐标存在 θ 的夹角,对该轨迹进行时间分割插补后可以得到三个插补点 p_1, p_2, p_3 。机械手从轨迹的第一个插补点 $p_1(x_1, y_1)$ 以速度 $v_1(v_{1x}, v_{1y})$ 按照预先设定的轨迹运动到第二个插补点 $p_2(x_2, y_2)$,在一个插补周期后,即在 p_2

(x_2, y_2) 点处进行实时位置反馈。此时可以发现由于Y轴负载的变化导致Y轴的速度比实际规划的速度要小,所以在一个插补周期后导致机械手未到指定的点 p_2 上,而是在点 p_2^* 上。最后产生的位置误差是 $p_\Delta(x_\Delta, y_\Delta)$,

$$\begin{aligned} [0055] \quad x_\Delta &= x_2 - x_2^* \\ y_\Delta &= y_2 - y_2^* \end{aligned} \quad (1)$$

[0056] 同时可以通过计算得到实际的X轴和Y轴的速度 $V_1^*(v_{1x}^*, v_{1y}^*)$,对比实际的速度和规划的速度可以比较各自轴的速度快慢。假设图中所示的实际速度 V_1^* 与规划速度 V_{1y} 有较大的差距,则将Y轴作为下次速度规划参考轴,这时的点 $p_2(x_2, y_2)$ 的速度 $v_2(v_{2x}, v_{2y})$:

$$[0057] \quad \begin{cases} v_{2x} = (v_{2y} * T - x_\Delta) / (T * \tan \theta) \\ v_{2y} = v_{1y}^* \end{cases} \quad (2)$$

[0058] 从而得出插补点 p_2 的速度,结合插补周期 T ,可以得到新的插补点 p_m ,这在理论上消除了由于负载变化导致的轨迹误差,增加了机械手轨迹控制的准确性。从而得出插补点 p_2 的速度,结合插补周期 T ,可以得到新的插补点 p_m ,这在理论上消除了由于负载变化导致的轨迹误差,增加了机械手轨迹控制的准确性。

[0059] 参照图5,该图为电机控制流程图。

[0060] 在本发明中,CoDeSys软件平台可以对伺服电机进行相应的参数配置。首先要对伺服电机的各参数进行初始化,然后通过CoDeS-ys IDE界面下的设备树进行设备在线扫描,这样可以将关节电机显示在设备树下的界面中。若扫描到,可以通过设置此关节电机对应的CFC功能块进行使能设置。然后通过预先编写好的算法获取笛卡尔空间轨迹,并且求出各个时刻关机理想的速度和执行情况。此时通过实时速度规划算法获取当前的实际关节速度并进行实时位置反馈,通过比较计算下一时刻的速度补偿量。如此通过反馈可以在一定范围内消除位置误差,这样提高了机械手的控制精度和实时性能。

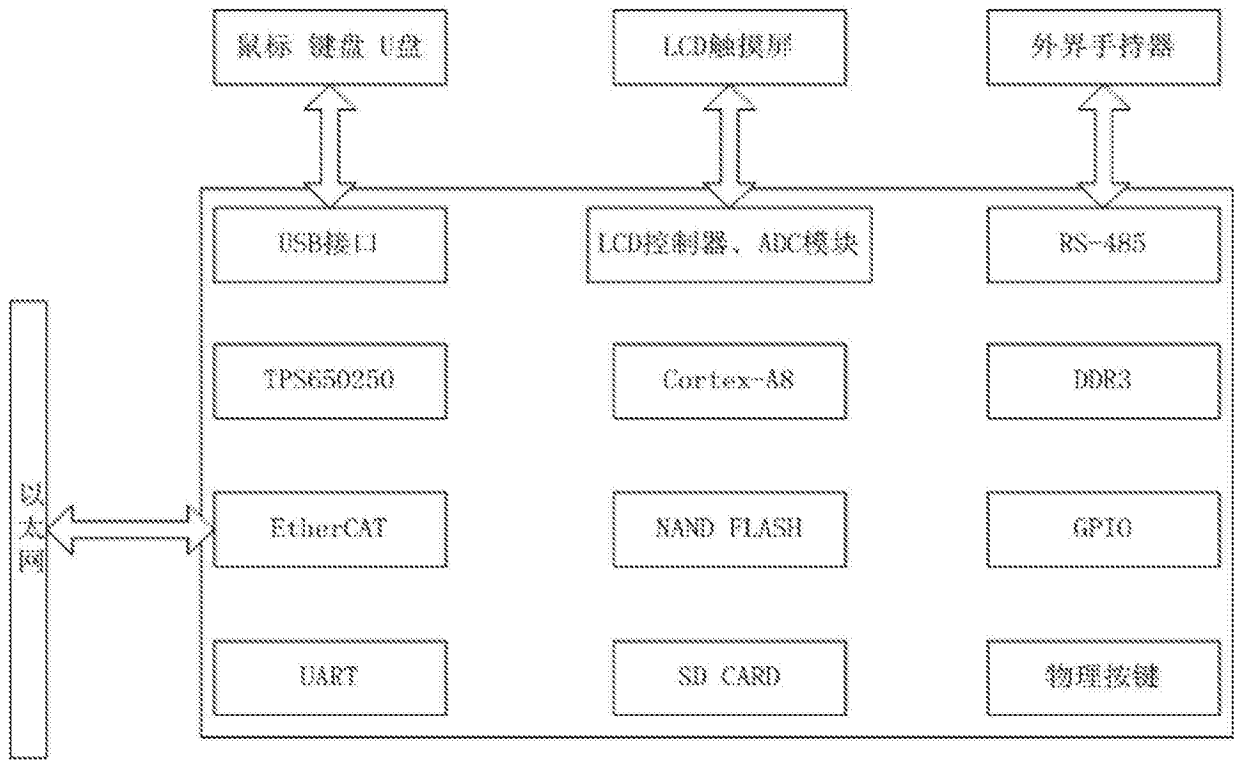


图1

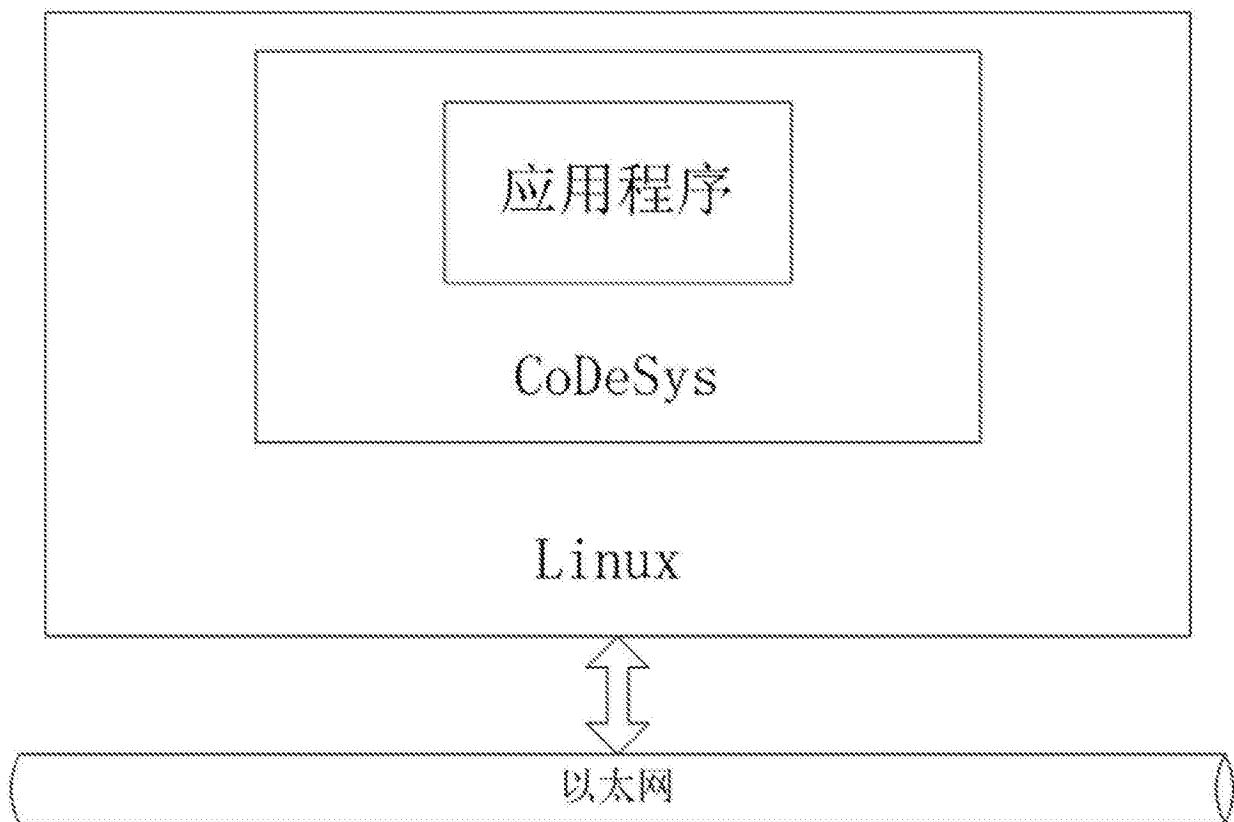


图2

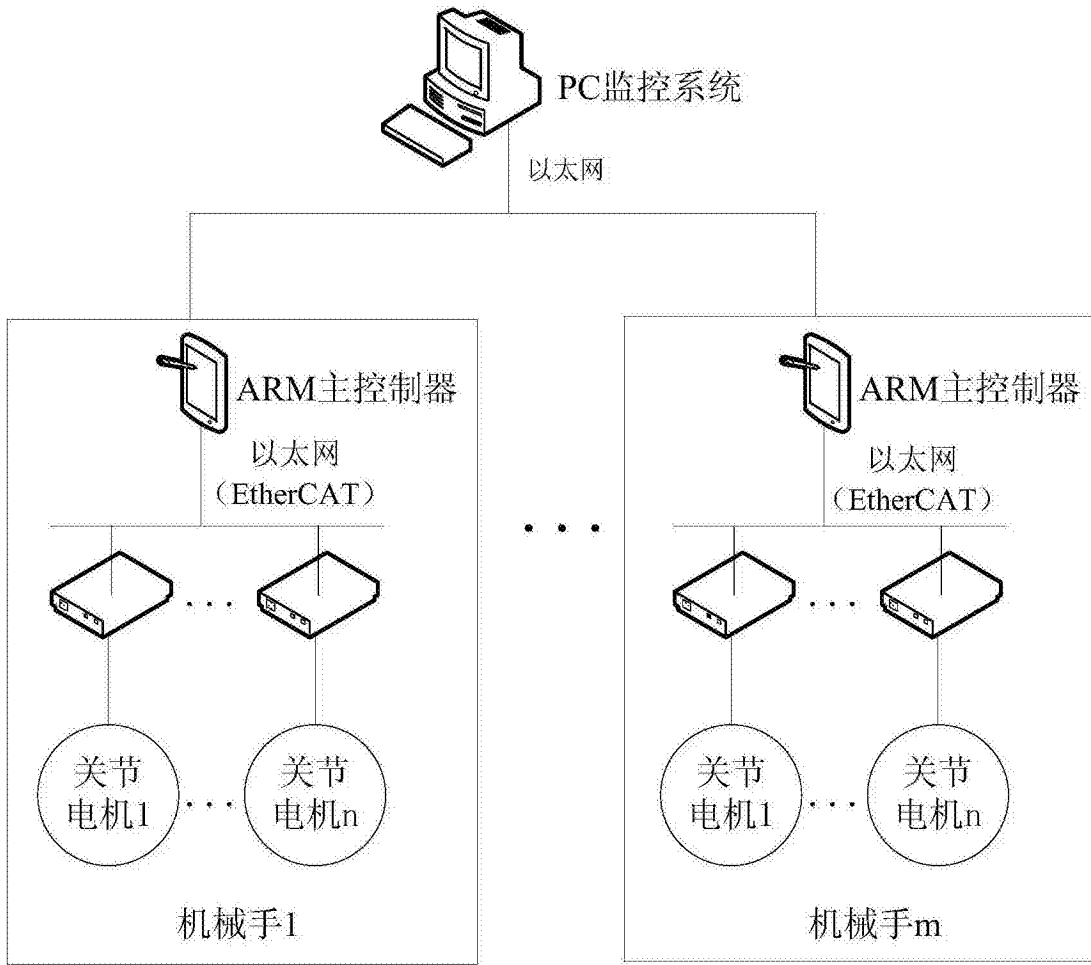


图3

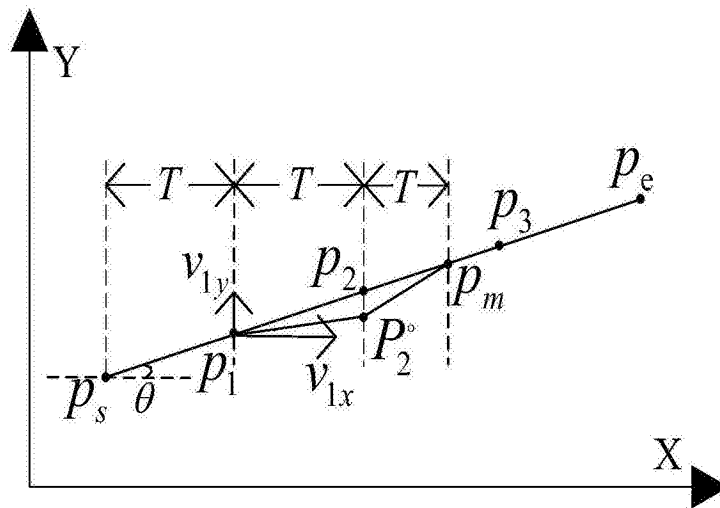


图4

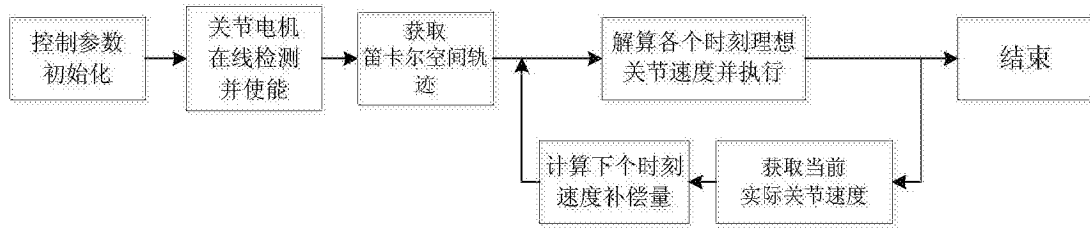


图5