

基于 PLCopen 的六轴工业机器人运动控制功能块设计

徐胜华, 徐建明, 赵 帅

(浙江工业大学 信息工程学院, 杭州 310023)

摘要: 针对工业机器人控制系统的开放性要求, 设计符合 PLCopen 规范的六轴工业机器人运动控制功能块; 基于 CodeSys 软件开发环境, 结合 SMC_CNC 库中数组、结构体和功能块, 采用 IEC61131-3 编程语言开发了六轴工业机器人运动控制功能块, 包含: 坐标变换、正逆运动变换、直线与圆弧插补运动等; 在此基础上, 开发了一系列机器人运动控制指令和可视化编程界面; 最后, 以埃夫特 ER50 六轴工业机器人为实验对象, 采用研制的机器人运动控制器, 实施机器人编程控制实验, 结果表明所开发的机器人运动控制功能块及其指令达到了设计要求。

关键词: PLCopen; 运动控制; CodeSys; IEC61131-3; 工业机器人

Design of Motion Control Function Blocks for six-axis Industrial Robot Based on PLCopen

Xu Shenghua, Xu Jianming, Zhao Shuai

(College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023, China)

Abstract: Aiming at the open requirements of industrial robot control systems, according to PLCopen specification, motion control function blocks are designed for six-axis industrial robots. Based on CodeSys software development environment, combined with the arrays, structures and function blocks in the SMC_CNC library, a few of motion control function blocks are developed using IEC61131-3 programming languages for a six-axis industrial robot, including coordinate transformation, forward and reverse kinematics transformation, linear and circular interpolation motion, etc. On this basis, a series of robotic motion control commands and visual programming interfaces are developed. Finally, taking the Efort ER50 six-axis industrial robot as experimental objects, and adopting to the developed robotic motion controller, robotic programming control experiments are implemented. The results show that the developed motion control function blocks and commands met the design requirements.

Keywords: PLCopen; motion control; CodeSys; IEC61131-3; industrial robot

0 引言

近几年来, 随着德国“工业 4.0”和“中国制造 2025”的先后提出, “智能制造”已经成为全球工业发展新的代名词。工业机器人作为智能制造的重要组成部分, 将在“智慧工厂”、“智能物流”、“全自动化流水线”等智能制造领域扮演着重要角色。而运动控制技术作为工业机器人的核心技术, 直接影响着工业机器人技术的发展, 因此, 发展和推进工业机器人运动控制技术具有十分重要的实践意义^[1-3]。

目前, 工业机器人的运动控制基本分为两种方式。一是采用专用的控制器, 专用的编程语言, 即采用原始设备

制造商 (original equipment manufactures, OEM) 方式。各个机器人厂商有自己专用的编程语言^[4-5]。例如, 瑞士的 ABB 工业机器人采用 RAPID 编程语言, Säubli 机器人使用 VAL3 语言, 德国的库卡机器人采用 KRL 编程语言, 日本的安川机器人采用 INFORM 语言, 发那科机器人则使用 Karel 编程语言。

另一种是基于 PLC 的运动控制方式。采用符合 PLCopen 标准的运动控制功能块作为编程控制基础。这种控制方式不受各种软硬件平台的限制, 具有开放性、易编程性、可移植性、通用性和可扩展性等特点^[4-5]。如 3S 公司的 CodeSys、Beckhoff 公司的 TwinCAT、KW 公司的 Multi-Prog、固高公司的 OtoStudio 等开发环境均是基于此种编程方式开发的。目前, 南京航空航天大学张得礼教授团队采用工控机+PMAC 运动控制卡的架构在 Codesys 环境下基于 PLCopen 轴组方式对运动控制功能块进行研究与开发^[6-7], 华中科技大学唐小琦教授团队结合 IEC61131-3 语言及 PLCopen 标准, 基于 Beremiz 开源平台研发了应用于数控机床控制系统的运动控制功能块^[8-9]。华南理工大学李

收稿日期: 2018-09-07; **修回日期:** 2018-09-29。

基金项目: 国家自然科学基金面上项目 (61374103); 浙江省自然科学基金重点项目 (LZ15F030003)。

作者简介: 徐胜华 (1994-), 男, 湖南岳阳人, 硕士研究生, 主要从事工业机器人和运动控制技术方向的研究。

徐建明 (1970-), 男, 江西吉安人, 教授, 博士研究生导师, 主要从事工业机器人和伺服控制技术方向的研究。

迪教授团队采用 DSP+FPGA 为架构的方案基于 PLCopen 对单轴及多轴运动控制功能块进行了研究^[10-12]。

基于以上两种运动控制编程方式的各自特点, 本文选择采用通用性和可扩展性更好的基于 PLCopen 标准的运动控制编程方式, 采用 ARM+Linux+RunTime System 的硬件控制方案, 以 3S 公司的 CodeSys 作为机器人控制系统的软件开发环境, 以 PLCopen 标准的运动控制功能块为基础, 结合 SoftMotionCNC 运动控制库开发出六轴工业机器人运动控制功能块, 包含: 坐标变换、正逆运动变换、直线与圆弧插补运动等, 并在此基础上开发出系列运动控制指令, 包括运动控制类、位置偏移类、回零、急停和复位等指令, 同时设计出用于人机交互的可视化编程控制界面。最后, 以埃夫特 ER50 机器人为实验对象, 采用研制的机器人控制器, 对开发的运动控制功能块及指令进行实验验证, 结果表明, 功能块及指令执行正确, 满足六轴工业机器人运动控制的要求。

1 控制系统结构方案

整个六轴工业机器人控制系统采用一主多从的控制模式, 主站采用 ARM+Linux 控制方案, 从站采用带 EtherCAT 接口的清能德创 CoolA8 系列伺服驱动, 主站与从站、从站与从站之间采用工业以太网 EtherCAT 技术进行通讯, 软件平台以 Codesys 3.5sp10 作为开发环境, 以 ST 和 CFC 作为主要编程语言进行机器人运动控制功能块的开发以及机器人控制算法的开发, 机器人控制系统结构图如图 1 所示。

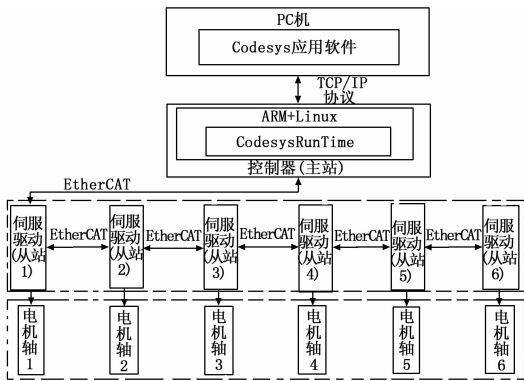


图 1 机器人控制系统结构图

PLCopen 组织自成立以来共发布了六个部分控制标准, 分别为运动控制库、扩展、用户导则、运动控制协调、回零功能、液压驱动扩展。这六个部分构成了 PLCopen 运动控制的基础。第一部分主要介绍了定义的运动控制功能块库, 包括单轴功能块和部分多轴功能块, 并定义了功能块的状态图, 第二部分对第一部分的功能块库进行了补充, 添加了部分单轴功能块, 并进一步完善了功能块的状态图, 第三部分列举了数个使用 PLCopen 功能块做控制系统的实例, 并对其做了详细说明, 第四部分介绍了轴组功能块及

其状态图, 第五部分详细介绍了单个运动轴的回零原理及常见回零方式, 并定义了部分回零功能块, 第六部分介绍了流体液压驱动设备功能块。

本文以一、二、四部分功能块为基础, 结合 SMC_CNC 库中的数组、结构体和功能块, 开发六轴工业机器人运动控制功能块及系列运动控制指令, 通过从可视化界面输入控制指令调用运动控制功能块达到控制机器人运动的目的, 最后, 在 CodeSys 开发环境中开发可视化编程界面, 用于实现指令的输入和对机器人运动控制的操作, 其软件结构框图如图 2 所示。

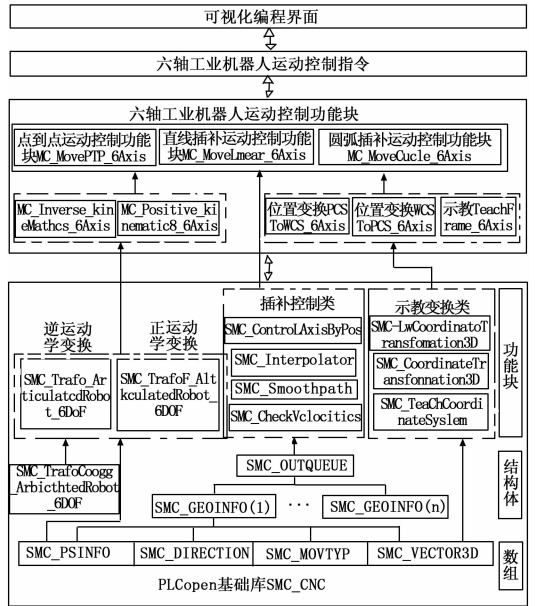


图 2 机器人运动控制软件结构图

由图 2 可知, 机器人运动控制软件结构可分为四层, 包括 PLCopen 基础库 SMC_CNC、六轴工业机器人运动控制功能块、机器人运动控制指令以及可视化编程界面。基础库 SMC_CNC 中, 包含数组、结构体和功能块, 其中, 结构体 SMC_OUTQUEUE 包含多个路径对象结构体 SMC_GEOINFO、SMC_GEOINFO 包含四个数组, 分别为 SMC_POSINFO、SMC_DIRECTION、SMC_MOVTYP、SMC_VECTOR3D, 库中的功能块对结构体和数组进行调用, 如示教变换类功能块调用 SMC_VECTOR3D, 插补控制类功能块调用结构体 SMC_OUTQUEUE, 正逆运动学变换功能块调用结构 SMC_Trafo_Articulated Robot_6DOF 数组 SMC_POSINFO。六轴工业机器人功能块层

包含位置变换类、正逆运动变换类、直线和圆弧插补类运动控制功能块。机器人运动控制指令层包含运动控制类、位置偏移类、回零、复位等指令, 通过对上层输入的文本解析, 并与指令系统中的指令匹配后, 调用下层对应的功能块实现对机器人的控制。可视化编程界面层用于指令程序的输入和人机交互。

2 运动控制功能块的设计

在 CodeSys 开发环境下, 添加 SMC_CNC 功能块库到库管理器中, 作为设计六轴工业机器人运动控制功能块的基础库, 库中常用数组、结构体如表 1 所示, 基础功能块如表 2 所示。

表 1 SMC_CNC 库数据结构

数组		结构体	
SMC_VECTOR3D	坐标系旋转矩阵	SMC_GEOINFO	路径对象结构体
SMC_MOVTYP	插补路径类型	SMC_OUTQUEUE	存储路径对象单元
SMC_DIRECTION	运动方向类型	SMC_POSINFO	存储位置信息结构体
SMC_INT_VELMODE	速度变化类型	SMC_Trafo_ArticulatedRobot_6DOF	六轴机器人 D-H 参数结构体

本文以上述数组、结构体、功能块为基础, 开发了符合 PLCopen 标准的、适用于六轴工业机器人运动控制的功能块, 如表 3 所示, 其中 PCS 表示工件坐标系, WCS 表示世界坐标系。

表 2 SMC_CNC 库功能块

功能块			
SMC_TeachCoordinateSystem	坐标系示教	SMC_Trafo_ArticulatedRobot_6DOF	六轴机器人逆运动学变换
SMC_SmoothPath	平滑路径角度	SMC_TrafoF_ArticulatedRobot_6DOF	六轴机器人正运动学变换
SMC_CheckVelocities	路径速度平滑	SMC_InvCoordinateTransformation3D	坐标系逆变换
SMC_Interpolator	路径插补	SMC_CoordinateTransformation3D	坐标系正变换
SMC_ControlAxisByPos	位置控制轴运动		

表 3 六轴工业机器人功能块介绍

六轴工业机器人功能块			
MC_MovePTP_6Axis	点到点运动控制	TeachFrame_6Axis	坐标系示教
MC_MoveLinear_6Axis	直线插补运动控制	MC_Positive_kinematics_6Axis	六轴机器人正运动学变换
MC_MoveCircle_6Axis	圆弧插补运动控制		
WCSToPCS_6Axis	坐标变换 (PCS 到 WCS)	MC_Inverse_kinematics_6Axis	六轴机器人逆运动学变换

下面以点到点运动控制功能块和圆弧插补运动控制功能块为例详细介绍六轴工业机器人运动控制功能块的设计流程。

2.1 点到点运动控制功能块设计

机器人点到点运动控制功能块的作用是控制机器人的工具中心点 (tool center point, TCP) 在空间中以最快的速度从起始点运动到目标点, 它主要应用于不需考虑 TCP 运动的轨迹形状, 而要求在最短时间内到达的运动场合。设计的六轴工业机器人点到点运动控制功能块流程如图 3 所示。

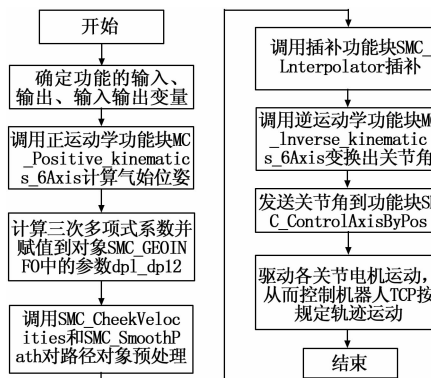


图 3 点到点运动控制功能块流程图

(1) 参照 PLCopen 功能块标准, 确定功能块的输入变量、输出变量以及输入输出变量, 然后对其进行定义和初始化, 变量如图 4 中接口所示。(2) 通过调用正运动学功能块 MC_Positive_kinematics_6Axis 计算出当前机器人 TCP 的位姿值, 将当前 TCP 的位姿作为机器人运动的起始点的位姿。接着根据起始位置和目标位置的数值, 以及初始和结束时机器人 TCP 的速度均为零等约束条件, 即可计算出机器人各轴运动的三次多项式函数的各项系数, 共 12 个数据。(3) 将前面计算出的起始位姿、三次多项式参数、输入的目标位姿、速度、加速度、减加速度等数据分别赋值到路径信息对象 SMC_GEOINFO 对应的参数 piStartPos、dP1—dP12、piDestPos、dVel、dAcel、dDecel 中, 完成之后调用 SMC_AppendObj() 函数将该对象添加到结构体 SMC_OUTQUEUE 中, 将其作为路径预处理对象。(4) 调用 SMC_CheckVelocities 功能块和 SMC_SmoothPath 功能块分别对路径对象的速度和位置进行平滑化处理, 使得运动轨迹更加平滑, 运动速度更加平稳。(5) 调用 SMC_Interpolator 功能块对预处理之后的路径对象进行路径插补, 通过插补功能块可以将初始位姿和目标位姿之间的路径插补出单独的位置点。(6) 调用设计的逆运动学功能块 MC_Inverse_kinematics_6Axis, 将 (5) 插补出来的位置点通过逆变换, 得到一组关节角。(7) 将关节角的值相对应的输入到功能块 SMC_ControlAxisByPos, 从而控制各个关节轴的运动, 最终实现点到点的插补运动, 到达目标点的位置。

下面通过一个实例来描述功能块在执行过程中 TCP 的位置和速度、加速度的变化, 以及功能块输入输出变量

的状态变化关系。调用两个点到点运动控制功能块，记为 FB1 和 FB2，将目标点位姿 P_1 (1200, 100, -1000, 40, 50, 60) 作为 FB1 的位姿输入、 P_2 (1400, 200, -1300, 70, 50, 30) 作为 FB2 的输入，然后将 FB1. bDone 与 FB2. bExecute 连接，使得功能块 FB1 执行完成后触发 FB2 的执行，从而实现机器人 TCP 从当前位置按点到点运动的方式运动到 P_1 点，再到 P_2 点。

通过使用 CodeSys 中数据跟踪插件对运动过程中变量数据的变化进行实时跟踪，程序运行中功能块变量状态时序图及速度、加速度、位置曲线如图 4 所示。

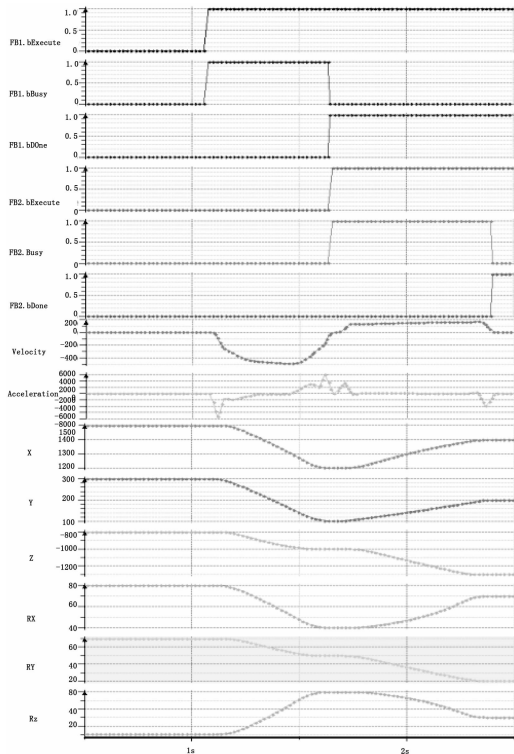


图 4 功能块各变量数据曲线

当 $t=1.2$ s 时，FB1. bExecute 由 0 变 1，功能块 FB1 开始执行，FB1. bBusy 状态为真，机器人开始加速运动，TCP 的位置开始变化；在 $t=1.7$ s 时，机器人到达第一个目标位置点 P_1 ，功能块 FB1 执行完成，此时 FB1. bBusy 状态由 1 变 0，FB1. bDone 状态由 0 变 1，使得 FB2. bExecute 为 1，FB2 开始执行，FB2. bBusy 为 1；直到 $t=2.4$ s 时，FB2. bDone 为 1，FB2. bBusy 由 1 变 0，机器人 TCP 到达第二个目标点 P_2 ，此时，速度、加速度等均减小至零，标志着功能块 FB2 执行完成。

2.2 圆弧插补运动控制功能块设计

机器人圆弧插补运动控制功能块的作用是控制机器人 TCP 在空间中根据给定的中间点和目标点，以圆弧插补轨迹方式从起始点经过中间点运动到目标点。根据空间中三点能唯一确定一个圆的数学原理，将机器人起始位置、输入到功能块的中间点位置和目标点位置坐标采用数学方法，

唯一确定机器人在空间运动的圆弧轨迹，经过插补算法，即可实现控制机器人 TCP 在空间中按照规划好的圆弧轨迹方式运动。圆弧插补功能块的流程如图 5 所示。

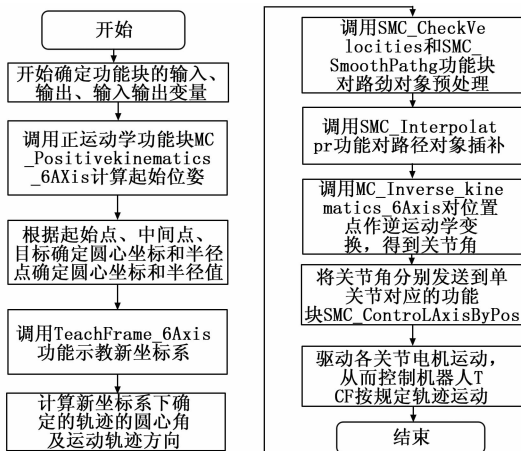


图 5 圆弧插补运动控制功能块流程图

(1) 定义功能块的输入变量、输出变量以及输入输出变量。由于在空间中三点才能唯一确定一个圆，所以不仅要有目标点作为输入，还要一个中间点作为输入，其他变量与点到点运动控制功能块保持一致。(2) 调用正运动学功能块 `MC_Positive_kinematics_6Axis`，计算出机器人 TCP 当前所处位置的位姿，作为机器人运动的起始点。(3) 根据输入的中间点位置和目标点位置的数值，确定圆弧的圆心和半径。由数学原理可知，空间三点可以构成唯一的一个圆，该圆也称三点连接的三角形的外接圆，而圆心就是该三角形的外心，半径则是圆心与任意一点的距离，根据数学公式可求得圆心和半径。(4) 调用三点示教功能块 `TeachFrame_6Axis`，以圆心为新坐标系原点，圆心到起始点为 ox 轴方向，圆心、起始点、中间点、目标点所在平面为 xoy 平面，建立新的坐标系。(5) 将新建立的坐标系记为 PCS，则调用不同坐标系下位置坐标转换功能块 `WCS-ToPCS_6Axis`，可以确定起始点、中间点、目标点在新坐标系下的坐标值。(6) 根据起始点、中间点、目标点在新坐标系下的坐标值，可以确定其在新坐标系下的位置关系，从而确定圆弧轨迹的方向是顺时针还是逆时针，根据运动方向，再确定圆弧所对应的圆心角的大小，即圆弧运动的旋转角度。(7) 将以上步骤求得的圆心坐标、半径值、轨迹方向、旋转角度分别赋值给路径对象 `SMC_GEOINFO` 中的参数 ($dP1$ 、 $dP2$ 、 $dP4$)、 $dP3$ 、 $iMoveType$ 、 $dT2$ ，将输入的速度、加速度、减加速度参数分别赋值给 $dVel$ 、 $dAccel$ 、 $dDecel$ ，将示教得出的坐标系列向量分别赋值给 Vx 、 Vy 、 Vz ，完成之后调用 `SMC_AppendObj()` 函数将该对象添加到结构体 `SMC_OUTQUEUE` 中，然后将其作为路径预处理对象。接下来的步骤与点到点运动控制功能块的设计步骤(4)~(7)完全一致，这样整个圆弧插补运

动控制功能块的设计流程基本完成。

通过实例验证设计的圆弧插补功能块在控制机器人运动过程中, 各输入输出状态变量的变化情况, 以及位置、速度、加速度等变化关系。通过调用两个圆弧插补运动控制功能块, 分别为 FB1 和 FB2, 将 FB1 的执行完成输出变量 bDone 连接到 FB2 的输入行变量 bExecute, 当 FB1 执行完后, 触发 FB2 的执行, 从而控制机器人完成两段圆弧轨迹的运行。

通过使用 Codesys 中数据跟踪插件对运动过程中各变量数据进行实时跟踪, 得出各功能块输入输出变量的时序图和位置等变量的曲线如图 6 所示, 在 $t=1\text{ s}$ 时, FB1. bExecute 触发, FB1 开始执行, FB1. bBusy 由 0 变 1, 机器人开始运动, 速度、加速度, 位置开始变化; 在 $t=14\text{ s}$ 时, 机器人运动到第一个目标点 P1, 标志着功能块 FB1 执行完成, FB1. bDone 变为 1, 同时触发 FB2 执行, 此时 FB2. bBusy 由 0 变为 1, 速度、加速度等再次从零变为正, 机器人开始加速运动; 在 $t=20\text{ s}$ 时, 机器人运动到终点 P2, FB2. bDone 由 0 变为 1, 表示功能块 FB2 执行完成; 在 $t=22\text{ s}$ 时刻, 将 FB1. bExecute 复位, 所有变量全部复位为 0, 功能块被重新初始化。

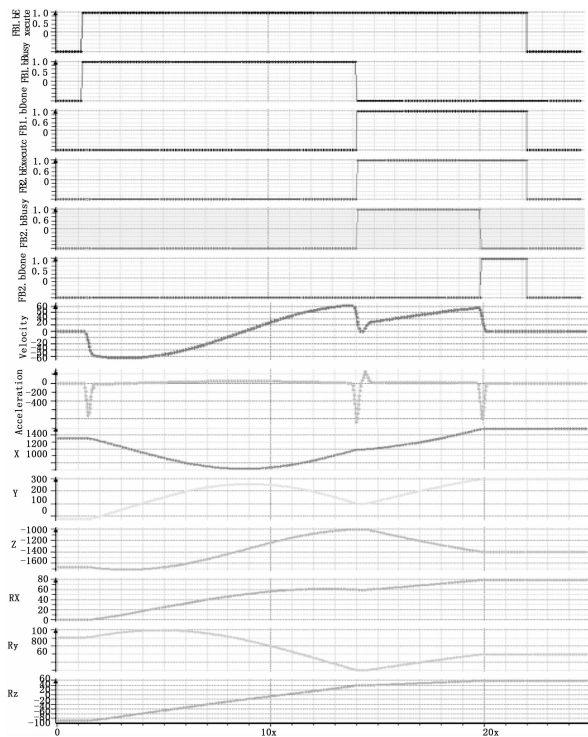


图 6 功能块各变量变化曲线图

3 运动控制指令及可视化编程界面

开发机器人的运动控制指令及可视化编程界面是为了能够直接而简易的控制机器人的运动, 同时也是为了能够验证设计的运动控制功能块的可行性。通过在编程界面上输入控制指令, 在指令解析程序中将指令进行解析, 调用

开发的机器人运动控制功能块, 实现对机器人的运动控制。

3.1 运动控制指令设计

机器人运动控制指令主要包括控制机器人进行点对点运动、直线插补运动、圆弧插补运动、TCP 位置偏移、坐标系变换、回原点等。通过对机器人功能需求进行分析, 开发了如表 4 所示控制指令。

表 4 运动控制指令及功能

名称	功能	名称	功能
movej	点对点运动	delay	延时
movel	直线插补运动	coord	坐标系切换
movec	圆弧插补运动	reset	机器人复位
ptprel	点到点偏移	home	回到原点
linrel	直线偏移	stop	停止运动

首先, 通过调用 CodeSys 开发环境下的文件读取函数 SysFileRead (), 读取从编程界面输入到文本框中的指令字符, 然后将字符以 ASCII 码的形式存入到声明的指定内存中, 通过指令解析程序, 将内存中的 ASCII 码与设计的指令的 ASCII 码一一匹配, 一旦与设计的指令匹配成功, 则调用指令对应的运动控制功能块, 将指令后面的位置参数赋值给功能块的输入位置变量, 然后执行该功能块, 从而控制机器人运动到指定位置。

以 movej 指令为例, 首先调用 SysFileRead () 函数, 读取编程界面文件中的字符, 以 ASCII 码形式存入内存 Buffer 中, 调用指令解析程序, 将内存中的 ASCII 码与 movej 每一个字符对应的 ASCII 码匹配, 若匹配成功, 则将指令后面括号中的参数赋值给点到点运动控制功能块的目标位置输入值 fDesPos_dX、fDesPos_dY、fDesPos_dZ 等, 然后执行功能块 MC_MovePTP_6Axis, 控制机器人执行点到点运动。

3.2 可视化编程界面

为了通过实验验证设计的运动控制指令及运动控制功能块的可行性, 开发了可视化编程界面, 该界面由文本输入框、文本显示框、按键等可视化元素排列组成。文本输入框用于运动控制指令程序的输入, 通过调用 SysFileRead () 函数, 读取指令程序字符, 用于程序解析。界面右上角五个按键用于对文本文件的操作, 分别关联到文本输入框的 newfile、openfile、savefile、closefile、deletefile 变量, 右下角为文件显示栏, 显示控制机器人控制程序的文件名。底部五个按键中, 编译按键关联指令解析程序的开始执行变量 start, 运行按键关联指令程序执行变量 run, 用于执行指令程序控制机器人运动, 停止按键关联程序停止变量 stop, 用于停止机器人运动。示教和返回按键用于切换可视化显示界面。

4 实验验证

以埃夫特 ER50 六轴工业机器人为实验对象，验证六轴工业机器人运动控制功能块及运动控制指令功能。ER50 机器人的 D-H 参数表如表 5 所示，依照 ER50 机器人模型建立连杆坐标系模型，如图 7 所示。

表 5 ER50 机器人 D-H 参数表

序号	关节偏角 (θ_i)	连杆偏距 (d_i)	连杆长度 (a_{i-1})	连杆转角 (α_{i-1})
1	0	-563	220	-90
2	90	0	900	0
3	0	0	160	90
4	0	1014	0	90
5	0	0	0	-90
6	0	200	0	0

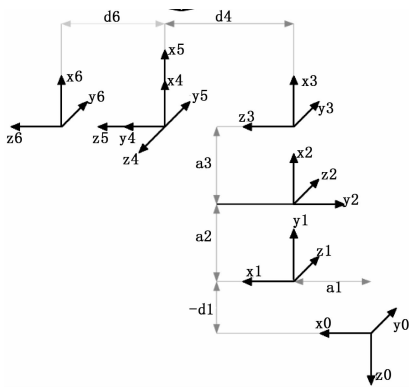


图 7 六轴机器人连杆坐标系

通过在可视化编程界面上输入指令程序 movej (1200, 100, -1000, 30, 40, 50); movel (1400, 300, -1500, 60, 30, 20); movec (1000, 200, -1000, 30, 50, 60, 1200, 100, -1200, 50, 20, 30); 即图 7 界面中的程序段来控制机器人运动，通过在 CodeSys 开发环境下记录的 TCP 位置信息数据通过 Matlab 合成的三维运动轨迹如图 8 所示。

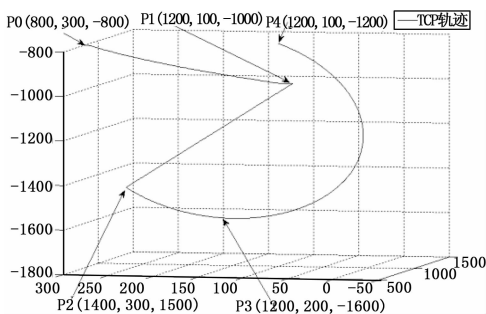


图 8 TCP 运动轨迹合成曲线图

如图 8 所示，首先，机器人 TCP 以点到点方式从 P₀ 运动至 P₁ 点，接着以直线插补方式运动至 P₂ 点，最后机器人

开始以圆弧插补的方式运动，到达 P₄ 点，轨迹曲线跟设计的程序控制机器人的功能完全一致。通过对机器人 TCP 的运动轨迹曲线图的分析，验证了设计的运动控制功能块及控制指令对机器人的控制功能。

5 结束语

本文在研究 PLCopen 运动控制功能块的基础上，在 CodeSys 软件开发环境下，采用符合 IEC61131-3 编程语言标准的 ST 和 CFC 编程语言，针对六轴工业机器人设计了系列运动控制功能块，设计了系列运动控制指令，调用运动控制功能块，实施对机器人的控制，并开发了可视化编程界面。以 ER50 六轴工业机器人作为实验对象，搭建了运动控制系统测试实验平台，在可视化编程界面输入运动控制指令，调用运动控制功能块，实施对机器人的运动控制，通过机器人 TCP 轨迹运动控制实验验证了运动控制功能块能够对六轴工业机器人实现运动控制，达到了设计的目的，同时该运动控制功能块具有良好的通用性和可扩展性，具有重要的应用价值。

参考文献:

- [1] 彭 瑜. 支撑智能制造的 PLCopen 规范和技术 [J]. 自动化仪表, 2015 (10).
- [2] 彭 瑜. 工业 4.0 基本单元开发研究中 PLC open 的作用和贡献 [J]. 机器视觉与智能控制, 2017 (6), 21-25.
- [3] 彭 瑜. 工控编程语言国际标准 IEC61131-3 及其影响 [J]. 2006 (4): 53-54.
- [4] Wicks M. PLC-based robotic controls versus OEM robotic controls: what's the best choice for your application, white paper [Z]. Intelli grated Inc. USA, 2011.
- [5] Dai F, A PLCopen-based approach for utilizing powerful industrial robot functions in PLC-controlled applications [M]. Springer International Publishing, Switzerland, 2013: 547-557.
- [6] 周 伟. 基于 PLCopen 的逻辑与运动控制功能块研究与开发 [D]. 南京: 南京航空航天大学. 2016.
- [7] 王夫康. PLCopen 轴组运动功能块的研究与实现 [J]. 机电工程, 2016 (12): 1478-1482.
- [8] 王 翰. 基于 PLCopen 的运动控制系统的设计 [J]. 新型工业化, 2013 (3): 69-70.
- [9] 王 翰. 基于 PLCopen 规范的运动控制系统研究与实现 [D]. 武汉: 华中科技大学. 2014.
- [10] 郑仲谦. 基于 PLCopen 的开放式运动控制器研究与实现 [D]. 广州: 华南理工大学. 2014.
- [11] 王翰博. 基于嵌入式软 PLC 技术的运动控制功能块的研究 [J]. 组合及机床与自动化加工技术, 2017 (2): 95-97.
- [12] 罗 亮. PLCopen 多轴圆弧运动控制功能块的实现 [J]. 自动化仪表, 2017 (2), 10-13.