

基于 CoDeSys 平台的六自由度工业机器人运动控制器设计

王耀东, 徐建明, 徐胜华

(浙江工业大学 信息工程学院, 杭州 310023)

摘要: 针对 ER50 六自由度工业机器人, 基于 CoDeSys 软件平台开发了一款机器人运动控制器; 采用 ARM+CoDeSys 架构和 PLCopen 规范进行六自由度工业机器人控制系统软件开发以及人机交互界面设计; 首先根据 D-H 参数对 ER50 机器人进行运动学建模, 并在此基础上封装 ER50 机器人正逆运动学功能块; 然后对控制器的示教模块、点动模块以及在线编程模块进行软件开发并设计各个模块的人机交互界面; 在 ER50 机器人上实现了运动控制器在线示教、点动以及在线编程等功能; 最后通过直线和圆弧轨迹跟踪实验验证了运动控制器设计的有效性。

关键词: 工业机器人; CoDeSys; PLCopen; 运动控制器

Design of 6 DOF Industrial Robot Motion Controller Based on CoDeSys Platform

Wang Yaodong, Xu Jianming, Xu Shenghua

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

Abstract: Aiming at ER50 industrial robot, a kind of robot motion controller is developed based on the CoDeSys software platform. Based on ARM+CoDeSys architecture and PLCopen specification, the 6 DOF industrial robot control system software and human-computer interaction interface are designed. Firstly, the ER50 robot kinematics is modeled according to the D-H parameters, and ER50 robot inverse kinematics block is packaged. Then, the teaching module, jogging module and online programming module are developed and the interactive interface of each module are designed. The motion controller online teaching and online programming function are achieved on the ER50 robot. Finally, the experiments of linear and arc trajectory tracking verify the effectiveness of motion controller.

Keywords: industrial robot; CoDeSys; PLCopen; motion controller

0 引言

随着机器人在现代工厂的广泛应用, 机器人技术得到了迅猛发展, 使得传统工业发生新的变革。在生产、医疗、物流等新兴领域, 机器人的应用改变了整个行业的发展模式。面对现代工业发展需求的不断提高, 降低生产成本、缩短开发周期和扩大应用领域尤为重要。目前, 机器人控制器多采用专用的控制器, 成本高, 开放性和移植性差。这就需要开发出一套高开放性、可移植性和可扩展性的机器人控制系统。

控制器是机器人控制系统的核心, 机器人的运动学求解、运动控制和轨迹规划等都由控制器完成。控制器大致

分为四类, 基于单片机的控制器^[1]、基于 PLC 的运动控制器^[2]、基于 PC 运动控制卡的控制器^[3]以及软件型运动控制器^[4]。其中基于 PCI 总线的运动控制卡控制方式和软件型运动控制器在当下运用最为广泛。运动控制卡采用高性能微处理器, 如 DSP 和大规模可编程设备, 因此具有比较强大的计算能力和逻辑处理能力, 大多数控制卡拥有统一的接口, 开放性强。而软件型运动控制器是建立在计算机强大的计算能力基础之上, 只需要将开发好的软件安装在 PC 或者嵌入式开发板上, 该平台就具有运动控制能力, 且该系统开放性强, 具有逻辑语言编程、CNC 多轴同步插补等优点。

目前, 任福深等人^[5]基于 PC 机开发了开放式的焊接机器人控制系统, 该系统运行稳定, 适应性强, 可以很好的完成焊接任务。马琼雄等人^[6]采用基于工控机 (IPC) 结合运动控制卡的结构形式, 并借鉴 ORC (Open Robot Controller) 等典型机器人控制系统软件体系结构设计了分层模块化的软件系统, 使得现有机器人控制系统只需要做少量修改就可以应用到不同场合。胡鹏等人^[7]采用 IPC+DSP 作

收稿日期: 2018-03-04; 修回日期: 2018-02-28。

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: 王耀东(1993-), 男, 安徽合肥人, 硕士研究生, 主要从事运动控制和迭代学习控制方向的研究。

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。

为六轴工业机器人的控制器，设计了一种基于可编程多轴控制器 PMAC (Programmable Multi-Axis Controller) 的开放式机器人控制系统，门昌华等人^[8]以 IPC+PMAC 作为 CIN CINNATI 工业机器人的控制器，设计了一种基于 PMAC 的开放式机器人控制系统。

本文以埃夫特公司的 ER50 机器人为研究对象，采用基于 ARM+CoDeSys 架构设计机器人运动控制器，并采用 EtherCAT 实时以太网通信技术，实时性强，运行速度快。该控制器具有机器人示教、在线编程、建立坐标系、关节点动控制、笛卡尔点动控制、轨迹插补规划、一键回零、各轴位置实时显示等功能。

1 控制系统结构方案

本文采用在 ARM 板上安装 CoDeSys 软件作为开发平台，构成一个支持逻辑与机器人运动控制相结合的可编程逻辑控制器。上位机采用 CoDeSys 软件提供的可视化模块进行设计开发，驱动器采用清能德创的 CoolA8 驱动器，控制总线选择 EtherCAT 通讯总线，机器人本体使用 ER50 机器人。整个系统采用模块化、总线式设计，因此整个机器人控制系统具有很强的通用性与开放性。系统架构如图 1 所示。

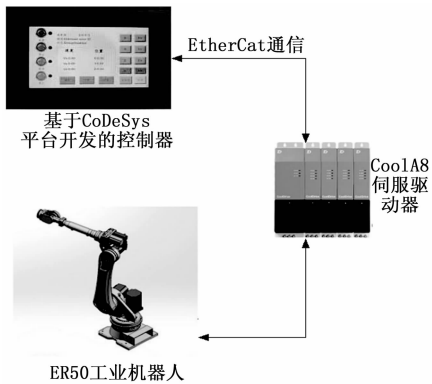


图 1 机器人控制系统整体架构图

本文在 CoDeSys 软件平台上，基于 PLCopen 规范^[9-10]对工业机器人运动控制器的软件部分进行设计。设计目标是基于 CoDeSys 准确搭建 ER50 机器人的正逆运动学模型，基于该模型开发一款控制精度高，运行稳定，且具有点动、示教、在线编程等功能的机器人运动控制器。首先对根据 ER50 机器人的 DH 参数建立连杆坐标系进而进行运动学建模，然后基于该运动学模型实现机器人关节空间点动，笛卡尔空间点动，基本的轨迹规划以及机器人示教和在线编程等功能，同时针对各功能设计良好的人机交互界面。

2 ER50 机器人运动学建模

机器人的机械结构和运动学密切相关。ER50 机器人的机械本体属于控制系统的下层，是控制系统上层机构的控制对象。机器人本体设计和动力学不在本文的研究范围之

内。机器人运动学方程又与本体结构密切相关，其运动学的正解和逆解是控制系统软件发送控制指令的依据，是工业机器人的轨迹规划、轨迹跟踪以及位置伺服控制的基础。因此本文以六自由度关节型机器人 ER50 为例，对工业机器人的运动学模型进行分析。

2.1 ER50 机器人连杆坐标系建立

对于 ER50 机器人相邻坐标系之间的坐标变换，一般先对全部连杆从基座到机器人末端执行器依次建立相邻连杆 $i-1$ 与 i 之间的变换关系。坐标系 $O_{i-1}X_{i-1}Y_{i-1}Z_{i-1}$ 的两次旋转和两次平移可以变换到坐标系 $O_iX_iY_iZ_i$ ，图 2 为相邻连杆坐标系的建立：

- 1) 绕轴 Z_{i-1} 旋转 θ_i 角，使得 X_{i-1} 轴与轴 X_i 处于同一个平面内；
- 2) 沿 Z_{i-1} 轴平移 d_i 个距离，使得轴 X_{i-1} 与轴 X_i 共线；
- 3) 沿 X_{i-1} 轴平移 a_i 个距离，使得坐标系 $O_{i-1}X_{i-1}Y_{i-1}Z_{i-1}$ 和 $O_iX_iY_iZ_i$ 坐标原点重合；
- 4) 绕 X_i 轴旋转 α_i 角，使两坐标系重合。

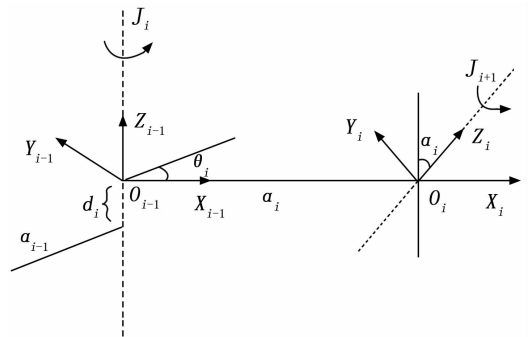


图 2 相邻连杆坐标系与参数

齐次变换矩阵可表示为：

$${}^{i-1}A_i = Rot(z, \theta_i) Trans(0, 0, d_i) Trans(a_i, 0, 0) Rot(x, \alpha_i) \tag{1}$$

其中： $Trans$ 为平移变换矩阵， Rot 为旋转变换矩阵，具体可表示为：

$${}^{i-1}A_i = \begin{bmatrix} \cos\theta_i & -\cos\alpha_i \sin\theta_i & \sin\alpha_i \sin\theta_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\alpha_i \cos\theta_i & -\sin\alpha_i \cos\theta_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

ER50 机器人的 D-H 参数如表 1 所示，其中 d_i 为关节轴 i 上的连杆偏距， a_{i-1} 为关节轴 $i-1$ 和关节轴 i 之间的公垂线长度（连杆长度）， α_{i-1} 表示关节轴 $i-1$ 和关节轴 i 之间连杆转角。由于机器人本体的机械传动耦合关系，四关节的转动将影响五、六关节的转动，五关节的转动又会带动六关节的旋转，关节耦合比如表 2 所示，本文设计的机器人运动控制器通过软件编程补偿了关节耦合对机器人控制产生的误差。由机器人的连杆参数和关节的旋转参数确定机器人操作臂末端坐标系在基坐标中位姿关系，建立 ER50

机器人连杆坐标系如图 3 所示, 根据该连杆坐标系关系建立 ER50 机器人的正逆运动学模型。

表 1 ER50 机器人 D-H 参数表

关节 (i)	变量 $\theta_i / (^\circ)$	$d_i /$ (mm)	$a_{i-1} /$ (mm)	$\alpha_{i-1} /$ (mm)	关节角 范围/ $(^\circ)$
1	θ_1	563	217	90	± 180
2	$\theta_2 (90)$	0	900	0	$-130-70$
3	θ_3	0	157	90	$-72-220$
4	θ_4	1017	0	90	± 360
5	θ_5	0	0	90	± 120
6	θ_6	195	0	0	± 360

表 2 ER50 机器人关节耦合比

关节 (i/i)	耦合比(角度)
4/5	$-1/68$
4/6	$-1/61$
5/6	$-1/61$

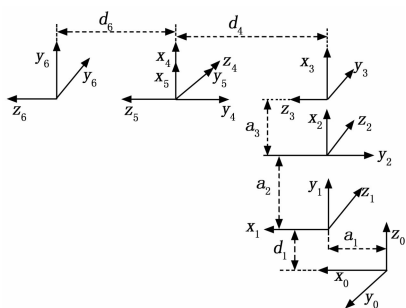


图 3 ER50 机器人连杆坐标系

2.2 ER50 机器人运动学模块封装

2.2.1 正运动学功能块封装

本文正逆运动学求解方式借鉴《机器人学导论》^[11] 书中所提出的方法。将正运动学公式用程序代码表述并封装成功能块, 程序功能块如图 4 (a) 所示, config 为机器人的 D-H 参数配置, drive0 至 drive5 为关节角当前位置信息, Position 是经过正运动学得出的笛卡尔空间位姿。armState、elbowState、handState 分别是第一关节、第三关节和第五关节的状态。

2.2.2 逆运动学功能块封装

机器人逆运动学是指已知机器人工具末端的位置和姿态, 通过逆运动学变换求出机器人各个关节轴对应的角度。由于本文设计对象为六轴串联型机器人, 所以根据逆运动学方程变换最多可求出八组逆解, 根据关节变化量最小的原则选取最优逆解, 逆运动学变换功能块在调用时的实例图如图 4 (b) 所示。模块中 _pre 表示上一次变换后各关节轴对应的关节角, pos 表示需要变换的机器人末端位姿。dq 表示经功能块变换后得出的各关节轴的角度。

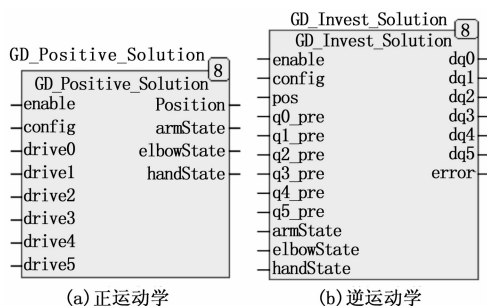


图 4 运动学功能封装

3 机器人运动控制器功能开发

在 CoDeSys 软件平台上开发机器人运动控制器的各个功能, 本文主要从示教模式、点动模式以及在线编程模式几个方面来阐述该运动控制器的软件开发过程。

3.1 示教模式

3.1.1 示教流程

通过使用笛卡尔空间点动或关节空间点动功能控制机器人工具末端运动到需要示教的位置点, 记录关节坐标以及笛卡尔空间末端执行器位姿信息。示教编程时, 可根据不同运动类型选择不同运动模式, 供选择的模式有 MoveL、MoveJ 以及 MoveC 三种运动模式, 在文件程序运行时判断指令的 ASCLL 码值, 然后再调用对应运动功能块。示教流程如图 6 所示。控制流程实现了 Wait () 函数, 采用定时器 TON 功能块实现, 定时时间可通过参数传入。

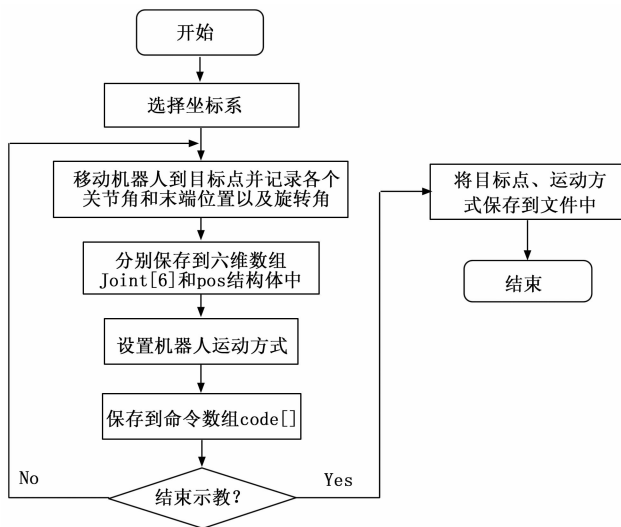
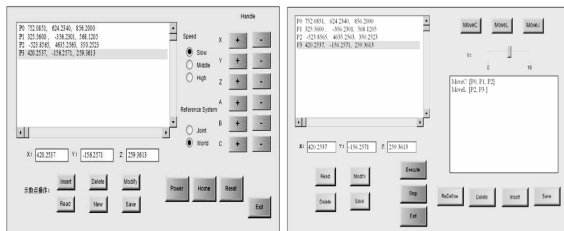


图 5 示教流程图

3.1.2 示教界面设计

手动操作机器人到各个示教点后保存示教点。再应用示教点指令生成示教程序, 保存程序到 .txt 文件中, 示教采集点和示教编程界面分别如图 6 (a)、6 (b) 所示。本系统根据机器人语言常用规范实现了一些基本指令, 组成本系统示教再现模块的机器人指令集。机器人的指令集包括

动作指令集、控制指令集。动作指令集包括 MoveC, MoveL, MoveJ, 控制指令集有 Wait、Jump。MoveC、MoveL 是圆弧插补和直线插补, MoveJ 是笛卡尔空间点到点的轨迹规划。在选择插补方式时设定对应运动速度 $v=5$, 执行语句 “MoveL [p_0, p_1];” 则机器人末端将从一个点 p_0 沿直线轨迹运动到当前点位置 p_1 , 其速度为 5 mm/s。通过 Read 按钮可以读取到示教文件上的各个示教点信息, 可以在线编辑修改示教文件, 以及插入、删除示教点。如果机器人在运行过程中出现问题, 为了安全保障, 可以选择 Stop 按钮, 紧急停止所有操作。



(a) 示教点采集界面 (b) 示教编程界面

图 6 示教界面

3.2 点动模式

3.2.1 关节空间点动

关节空间点动模式采用 MC_Jog 功能块实现, 可以单独控制关节轴的正反转, 以及运行时的速度、加速度, 也可以根据需要独立设置, Axis 为电机轴对应的控制字。机器人控制系统单轴模式的程序图如图 7 所示。MC_Power 功能块用于控制驱动器的使能状态, 是控制伺服驱动器电源模块, 上电使能是电机轴运动的必要前提。该模块中, Axis 为关节轴控制字, 对应于具体的关节轴电机驱动器, 通过置 Enable、bRegulatorOn 和 bDriveStart 三个输入控制字为 true 即可使能伺服驱动器电源。MC_Reset 实现电机轴的复位操作, 用于电机轴从 ErrorStop 状态到 StandStill 状态的转换。

3.2.2 笛卡尔空间点动

笛卡尔空间点动, 设计此功能的目的在于使用示教器示教或需要平移末端工具点时, 能更加快速到达目标点, 使用时更加方便和快捷。此功能设计有 6 个功能按键, 分别为 X+、X-、Y+、Y-、Z+、Z-, 分别控制机器人工具末端沿着世界坐标系中的 X、Y、Z 坐标轴移动。以 X 轴为例, 当点击 X+ 时, 工具末端会沿着 X 轴正方向移动。此功能的设计原理是通过功能按键来控制末端位置的坐标值增加或减小, 然后再根据机器人坐标系模型中的逆运动学方程, 将末端位置转换为各个关节角度, 再将转换出的各个关节角度作为各轴连接的电机的设定位置, 从而控制各轴同步运动到相应的位置, 达到预期目的。

3.2.3 点动模式界面设计

点动界面包含两种点动模式, 分别为关节空间点动和

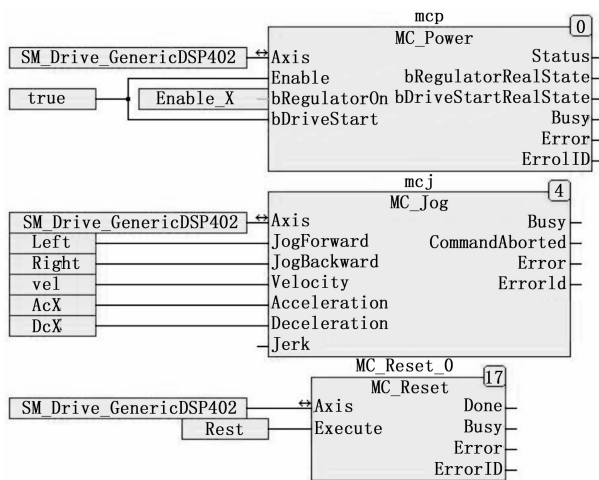
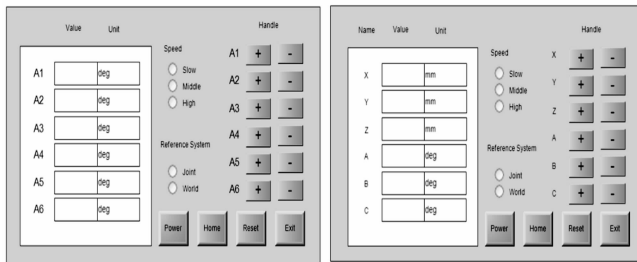


图 7 单轴模式程序图

笛卡尔点动, 每种模式单独设计一个界面, 界面包含点动控制的各功能按键、保障安全的急停按键、模式之间相互切换的切换按键、显示轴运动状态的状态显示灯、实时显示各轴关节位置或工具末端位姿数据的数据状态栏等。在关节空间下, 通过控制各个关节来实现手动运行如图 8 (a); 在运行速度上为了安全没有提供在线设置运行速度, 而是提供了三种运行速度模式, 分为 Slow、Middle、High 三种, 这样操作者可以根据需求在线选择运行速度。另一种是在笛卡尔空间进行手动操作如图 8 (b), 可以手动移动在空间中的位置以及姿态, 通过三维坐标的修改按钮, 以及欧拉角的旋转按钮进行增加欧拉角度或者减少, 从而改变末端姿态, 使其运行到期望位置。



(a) 关节空间点动界面 (b) 笛卡尔空间点动界面

图 8 点动界面

3.3 在线编程模式

3.3.1 在线编程实现

在线编程支持新建 .txt 编程文件, 或者加载之前已有的编程文件并再此基础上进行修改, 然后保存。运动指令主要有 MoveJ、MoveC 和 MoveL 三种, 当在线编程时目标位置以及中间点位置要在机器人运动范围内以及运动指令要符合特定语法, 如 MoveJ (10, 20, 30, 10, 10, 5, 5); 并以 “;” 结束, 前 6 个参数表示目标位姿, 最后一个参数表示运动速度。不符合这种语法在编译时会报错。错误类型主要有: 指令不符合规范、文件为空、指令没有以

分号结束等。编译完成无误情况下, 可以通过运行按钮执行文件中的指令。控制指令有 Wait 指令, 暂停一定时间。在线编程流程图如图 9 所示。

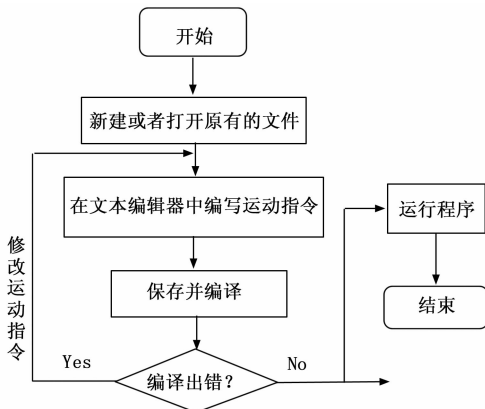


图 9 在线编程流程图

3.3.2 在线编程界面

在线编程界面如图 10 所示, 在线编程界面包含当前坐标系显示栏、机器人运动状态显示栏、程序编译错误显示栏、程序文件管理模块、程序文本编辑及修改栏、界面切换功能键、程序操作功能键及文件操作功能键。程序操作功能键包括对程序编译和运行, 文件操作功能键包括文件新建、打开、保存、关闭和删除。通过编程界面可在线管理程序代码, 存储在控制器中的程序文件也会在文件管理模块中显示, 也可以新建 .txt 类型的程序文件。通过对程序文件修改保存后, 可点击编译按钮, 对程序进行编译, 观察编写的程序是否有语法错误, 编译无误后, 可点击运行按钮, 机器人就会按照程序功能控制机器人按照预定的轨迹到达目标位置点。通过查看机器人状态显示栏查看机器人运动状态。



图 10 在线编程界面

4 实验结果与分析

在 ER50 六轴工业机器人上对设计的运动控制的在线示教、在线编程功能进行测试。实验对象如图 11。

使机器人姿态保持不变, 通过运动控制器的示教功能示教机器人位置, 保存 4 个点如下:

$$P_0(496.213, 170.285, 612.158, 16, 23, 121)$$

$$P_1(728.610, 296.926, 786.103, 16, 23, 121)$$



图 11 ER50 机器人本体+运动控制器

$$P_2(950, 196.734, 1723.02, 16, 23, 121)$$

$$P_3(1246.207, 287.654, 1810.143, 16, 23, 121)$$

再通过在线编程功能执行语句“MoveL [P₀, P₁];”实现点 P₀ 到 P₁ 的直线插补, 以及“MoveC [P₁, P₂, P₃];”实现点 P₁、P₂、P₃ 之间的圆弧插补。机器人末端位置的三维坐标记录如图 12 所示。将机器人运行数据保存起来并在 matlab 中对插值数据进行绘图, 得到规划的轨迹曲线图如图 13 所示, 将实际定位的 P₀、P₁、P₂、P₃ 点的坐标与示教定位点坐标进行误差对比, 数据如表 3 所示。

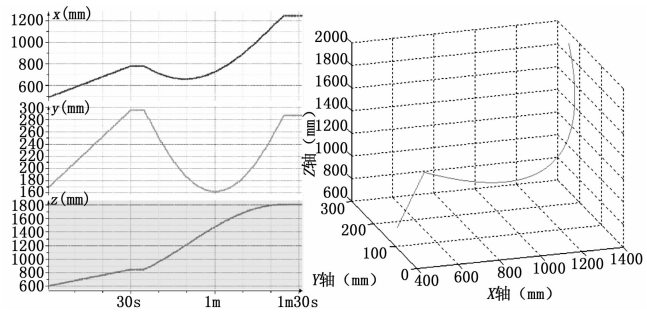


图 12 机器人末端坐标位置 图 13 机器人末端三维轨迹

表 3 定位精度表

坐标点	示教点坐标 (mm)	实际定位点 坐标(mm)	定位误差 (mm)
P ₀	$\begin{bmatrix} 496.213 \\ 170.285 \\ 612.158 \end{bmatrix}$	$\begin{bmatrix} 496.304 \\ 170.212 \\ 612.098 \end{bmatrix}$	0.1312
P ₁	$\begin{bmatrix} 728.610 \\ 296.926 \\ 786.103 \end{bmatrix}$	$\begin{bmatrix} 728.672 \\ 296.868 \\ 786.145 \end{bmatrix}$	0.0947
P ₂	$\begin{bmatrix} 950.000 \\ 196.734 \\ 1723.02 \end{bmatrix}$	$\begin{bmatrix} 950.112 \\ 196.685 \\ 1723.101 \end{bmatrix}$	0.1466
P ₃	$\begin{bmatrix} 1246.207 \\ 287.654 \\ 1810.143 \end{bmatrix}$	$\begin{bmatrix} 1246.317 \\ 287.725 \\ 1810.172 \end{bmatrix}$	0.1341