



(12)发明专利申请

(10)申请公布号 CN 106041928 A

(43)申请公布日 2016.10.26

(21)申请号 201610473462.4

(22)申请日 2016.06.24

(71)申请人 东南大学

地址 210096 江苏省南京市四牌楼2号

(72)发明人 甘亚辉 戴先中 邢继生 王政伟

郭哲

(74)专利代理机构 南京苏高专利商标事务所

(普通合伙) 32204

代理人 郑晶

(51) Int. Cl.

B25J 9/16(2006.01)

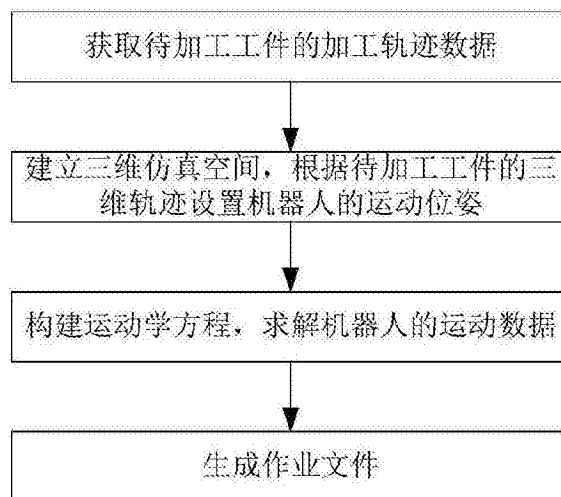
权利要求书2页 说明书6页 附图4页

(54)发明名称

一种基于工件模型的机器人作业任务生成方法

(57)摘要

本发明公开了一种基于工件模型的机器人作业任务生成方法,通过提取待加工工件的加工轨迹数据利用三维仿真空间显示机器人的运动轨迹,利用人机交互式手段设置机器人的运动轨迹和位姿数据,而后建立机器人运动学模型并根据位姿数据求解机器人运动数据,最终根据机器人运动数据生成机器人作业文件。本发明的机器人作业任务生成方法能实现工业机器人作业文件的快速生成,能对机器人和工件的模型进行逼真灵活地显示,所支持的机器人种类不受限制而且可以支持多机器人编程,还可以支持工业中常见的复杂曲线编程,能大大提高工业生产效率。



1. 一种基于工件模型的机器人作业任务生成方法,其特征在于,该方法包括以下步骤:

(1)获取待加工工件在数据交换文件中的加工轨迹数据,将该加工轨迹数据进行坐标系转换后再进行插补得到插补后的加工轨迹数据;

(2)建立三维仿真空间,使用虚拟现实建模语言建立待加工工件和机器人的三维实体模型并在三维仿真空间中显示,利用所述插补后的加工轨迹数据在三维仿真空间中显示待加工工件的三维轨迹,根据所述待加工工件的三维轨迹设置机器人的运动位姿;

(3)根据机器人的三维实体模型构建机器人的运动学方程,根据机器人的运动位姿利用运动学方程求解机器人的运动数据;

(4)根据所述机器人的运动数据生成作业文件。

2. 根据权利要求1所述的机器人作业任务生成方法,其特征在于,步骤(1)中待加工工件在在数据交换文件中的加工轨迹数据是指待加工工件的图形边界信息,图形边界信息包含不同类型的基本图元,若图元类型为点,则获取其对应坐标;若图元类型为圆弧,则获取圆心坐标、半径、起始角、终止角和方向向量;若图元类型为直线,则获取其起始点和终止点;若图元类型为圆,则获取其圆心坐标、半径和方向向量。

3. 根据权利要求2所述的机器人作业任务生成方法,其特征在于,步骤(1)对加工轨迹数据进行坐标转换是指将加工轨迹数据由对象坐标系转换为世界坐标系,即利用对象坐标系和世界坐标系之间的相对位置关系完成加工轨迹数据的坐标转换,确定对象坐标系和世界坐标系之间的相对位置关系,具体包括:

以任意单位长的向量作为对象坐标系的Z轴,利用任意轴算法确定对象坐标系的X轴,具体为:判断对象坐标系的Z轴是否靠近正或负的世界坐标系的X轴,若是,由世界坐标系的Y轴与对象坐标系的Z轴进行矢量积运算得到对象坐标系的X轴;否则,世界坐标系的Z轴与对象坐标系的Z轴进行矢量运算得到对象坐标系的X轴;所述靠近是指对象坐标系的Z轴在世界坐标系的X轴分量 Z_x 和Y轴分量 Z_y 同时小于 $1/64$;

再利用右手定则确定对象坐标系的Y轴。

4. 根据权利要求2所述的机器人作业任务生成方法,其特征在于,步骤(1)中对加工轨迹数据进行插补是指:若图元类型为直线,则采用等步长法进行插补,否则,采用等偏差法进行插补。

5. 根据权利要求1所述的机器人作业任务生成方法,其特征在于,所述运动位姿包括位置和姿态,位置通过目标轨迹来体现,目标轨迹的设置,包括以下步骤:

建立三维仿真空间中三维轨迹和windows平面上二维轨迹的转换关系;

提取用户鼠标在windows平面上的实时位置,并根据鼠标位置与二维轨迹的距离远近用不同颜色显示windows平面上的二维轨迹便于用户确定所选择的二维轨迹;

将用户选择的二维轨迹转换为三维仿真空间中的三维轨迹,获得机器人运动的目标轨迹,所述三维仿真空间采用开源软件库OpenGL实现。

6. 根据权利要求5所述的机器人作业任务生成方法,其特征在于,姿态的设置是指利用机器人最后一个关节的坐标系的x轴、y轴和z轴的方向来确定。

7. 根据权利要求1所述的机器人作业任务生成方法,其特征在于,步骤(3)中根据机器人的三维实体模型建立机器人的运动学方程是指:将机器人可以独立运动的连杆进行分离,在每个独立运动的连杆上建立坐标系,利用各连杆的旋转轴和旋转方向所确定的各坐

标系之间的转换关系。

一种基于工件模型的机器人作业任务生成方法

技术领域

[0001] 本发明涉及工业机器人技术领域,具体涉及一种基于工件模型的机器人作业任务生成方法。

背景技术

[0002] 机器人作业任务的生成,一般是从待加工工件的CAD模型获得机器人运动轨迹数据,然后利用智能编程由后台程序自动生成机器人作业文件。一般的思路为先建立仿真三维空间,用于显示工件和机器人的模型,同时提供用户设置机器人运动位姿的方式。现有的机器人作业任务生成方法多是采用一种直观的解决方案:使用成熟的三维CAD软件如AutoCAD、SolidWorks等软件提供的三维场景,并针对几种固定的机器人编写运动学算法求解机器人运动数据。然而,这种方案存在以下缺点:①支持的机器人种类非常有限;②CAD软件对计算机性能要求高,运行耗时长;③使用第三方平台开发会导致软件开放度不高。

发明内容

[0003] 发明目的:为了克服现有技术中存在的不足,本发明针对工业机器人的作业任务,提供一种基于工件模型的机器人作业任务生成方法,能够在人机交互性较强的三维仿真空间中提供一种更加高效和灵活的机器人作业任务。

[0004] 技术方案:为实现上述目的,本发明中基于工件模型的机器人作业任务生成方法,通过从待加工工件的CAD模型中提取加工轨迹数据,再利用三维仿真空间,根据加工轨迹数据设置机器人的运动位姿,再利用机器人的运动位姿数据通过机器人运动学方程求解获得机器人的运动数据,最后根据运动数据生成作业文件。具体包括以下步骤:

[0005] (1)获取工件模型中的加工轨迹数据

[0006] 本发明中采用的工件模型是CAD的数据交换文件,即DXF文件,由于DXF文件中的数据可以ASCII码形式呈现,采用高级程序语言C++对DXF文件中的实体段中的信息进行解析可以获得所需的加工轨迹数据。在获得的加工轨迹数据中有直线,圆和圆弧的原始数据,其中直线为二维图元,存在于世界坐标系中,不需要经过坐标系转换即可使用。三维图元(如圆和圆弧)存在于对象坐标系中,此类轨迹需要进行坐标系间的转换而生成世界坐标系中的数据。由于后续进行数据显示时采用了OpenGL提供的轨迹显示方式,所以需要世界坐标系WCS中的轨迹数据进行插补,以获得可以在世界坐标系中进行显示的轨迹类型。

[0007] (2)建立三维仿真空间,使用虚拟现实建模语言建立待加工工件和机器人的三维实体模型并在三维仿真空间中显示,利用所述插补后的加工轨迹数据在三维仿真空间中显示待加工工件的三维轨迹,根据所述待加工工件的三维轨迹设置机器人的运动位姿。

[0008] 在获得加工轨迹数据在世界坐标系WCS中的数据后,需要建立三维仿真环境以对轨迹进行显示,有利于高效设置机器人的运动轨迹。三维仿真空间是在计算机上生成作业文件的基础,仿真环境的首要任务显示仿真机器人和工件的三维实体模型以提供运动的实体和目标。三维场景模块是以开源软件库OpenGL为基础进行开发的,本发明中通过调用

OpenGL提供的一系列三维场景建立功能在计算机上生成三维场景,使用C++编写的程序不仅可以显示机器人和工件的三维实体模型,而且可以将获得的运动轨迹进行显示。本发明中采用虚拟现实建模语言(Virtual Reality Modeling Language,即VRML)对建立机器人和待加工工件的三维实体模型并进行渲染,VRML是一种标准的语言,其模型的各项参数均可用程序语言进行读取并显示;使用OpenGL中提供函数接口在三维场景中显示机器人的运动轨迹。

[0009] 运动位姿包括位置和姿态,其中位置通过目标轨迹体现,本发明中提出了一种灵活且高效地设置机器人运动的目标轨迹和姿态的方式。机器人运动目标轨迹的设置通过选择三维仿真环境中现有轨迹实现,但是由于计算机呈现的二维平面在选择三维空间中的轨迹时需要进行平面的切换,操作十分复杂。本发明中选择轨迹的方式具体实现为通过将OpenGL中的三维轨迹转换为Windows平面上的二维轨迹,然后,在程序中读取鼠标的实时位置,通过鼠标与二维轨迹的距离远近来给轨迹显示不同的颜色,用户可在轨迹颜色改变后点击鼠标从而确定选择该二维轨迹。最终,将二维轨迹转换为OpenGL中的三维轨迹,即可获得机器人运动的目标轨迹,这大大降低了用户选择轨迹的复杂度。

[0010] 对于目标姿态的设定,由于确定机器人末端坐标系的姿态即可确定机器人的姿态,因此本发明中主要通过设置机器人末端坐标系中x轴、y轴和z轴的方向来确定。本发明中提出了两种模式生成机器人姿态:自动生成和用户设置。第一种的实现方法是在用户确定所采用的机器人作业方式后即可确定姿态如何由多个曲面的约束关系生成,然后由预定的程序根据对应的约束关系自动计算出姿态的数据。第二种方法是通过用户的设置方式,此方法的主要思想是直接设置机器人末端坐标系的x轴和y轴的轴向来确定机器人姿态,用户可以通过选择三维仿真环境中对应的轨迹来确定x轴和y轴,用户所选择的轨迹的方向即为x轴或y轴 的方向。此外,用户还可以通过直接输入坐标系的x轴和y轴的方向的参数来确定,而z轴可以通过x轴和y轴通过常见的右手定则求得。至此,可以确定机器人运动的目标位姿。

[0011] (3)通过机器人三维实体模型建立运动学方程并求解机器人运动数据

[0012] 在本方法中采用关节型机器人的建模方法,此种方法的优点在于可以建立通用的运动学建模模块,通过用户载入的仿真机器人模型可以自动建立机器人的运动学模型,从而达到可以支持的机器人种类不受限制,并且可以完成多个机器人作业的任务。

[0013] 该种建模方法首先将机器人的各个可以独立运动的连杆分离,在每个连杆上建立坐标系,各坐标系用于确定连杆与上一连杆间的平移与旋转,在设定连杆的旋转轴和轴向后,可确定本连杆在旋转时的旋转中心轴和旋转方向,故可获得连杆*i*相对于上一个连杆*i-1*的坐标系关系 ${}^{i-1}T_i$,故可建立机器人学中的运动学方程为 ${}^0T_j = {}^0T_1 * {}^1T_2 * \dots * {}^{j-1}T_j$ (假设机器人由*j*个关节),由该方程可以求解机器人的运动数据。对于不同类型的机器人可以通过读取模型中的坐标轴数量确定机器人的轴数,从而建立不同机器人的运动学方程完成不同种类的机器人的运动数据求解,达到本发明中支持的机器人编程种类不受限制的效果。

[0014] (4)作业文件的自动编写

[0015] 在获得了运动数据,即机器人运动的类型及运动过程中关键点的各关节角数据后便可编写机器人作业文件,本发明中采用的作业文件的编写方式是使用高级程序语言编写的后台程序自动生成的方式,采用此种方式可以高效地完成机器人的作业任务,并且由于

不需要编程人员的参与,将有效地减少编程中的错误。此外,当加工轨迹数量较多或者轨迹为复杂轨迹时,编程的效率将明显提高且难度大大降低。

[0016] 该方式中首先需要确定机器人每条运动指令的类型,如直线运动或圆弧运动,运动指令的类型可以通过在步骤(2)中设置的目标轨迹类型进行确定。其次,需要确定该轨迹中运动的关键点处机器人各关节的数据,关键点是由机器人的运动轨迹所确定,对于直线运动,机器人需要确定的关键点为直线的两个端点,而对于圆弧运动,机器人需要确定的关键点为圆弧的起始点、终止点和任意一个中间点。利用用户设置的目标位姿和步骤(3)中的机器人运动学方程即可求得机器人在该关键点的各关节角。

[0017] 有益效果:本发明中基于工件模型的机器人作业任务生成方法,以工件模型中包含的轨迹数据为基础,根据用户设置的机器人目标轨迹确定运动类型,并根据用户所设置的机器人位姿和机器人运动学方程并计算出关键点数据得到机器人的运动数据,最后根据运动数据利用作业文件格式生成机器人的作业文件。相比于传统的作业任务生成方式,本发明机器人作业任务生成方法能够灵活且高效地设置机器人运动位姿,降低用户的工作量和难度,进一步降低作业文件的出错率,并且有效提高作业文件的精度;由于本发明的机器人作业任务生成方法采用开放式关节型机器人的建模方式构建运动学方程,所以支持的机器人种类不受限制,而且进行作业的机器人数量可由用户自由设置,具有极大的通用型和灵活性。

附图说明

[0018] 图1是本发明中基于工件模型的机器人作业任务生成方法的流程图;

[0019] 图2为DXF文件图元信息读取流程示意图;

[0020] 图3为DXF文件中的在世界坐标系和对象坐标系,图3(a)为世界坐标系,图3(b)为对象坐标系;

[0021] 图4为本发明中对加工轨迹数据进行插补的流程图;

[0022] 图5为本发明中用户设置机器人目标轨迹的示意图;

[0023] 图6为本发明中为建立通用关节型机器人模型的方式;

[0024] 图7为本发明中采用的机器人作业文件的格式。

具体实施方式

[0025] 下面结合实施例对本发明作更进一步的说明。

[0026] 本发明中基于工件模型的机器人作业任务生成方法,如图1所示,通过从待加工工件的CAD模型中提取加工轨迹数据,再利用三维仿真空间,根据加工轨迹数据设置机器人的目标轨迹,根据待加工工件的三维实体模型设置机器人的运动姿态,再利用机器人目标轨迹和运动位姿数据通过机器人运动学方程求解获得机器人的运动数据,最后根据运动数据生成作业文件。具体包括以下步骤:

[0027] (1)获取工件模型中的加工轨迹数据

[0028] 1)解析DXF文件中包含的加工轨迹数据

[0029] DXF是一种开放的矢量数据格式,DXF文件是由很多的“代码”和“值”组成的“数据对”构造而成,这里的代码称为“组码”(group code),指定其后的值的类型和用途。每个组

码和值必须为单独的一行的。DXF文件具有固定的格式,主要由HEADER段、CLASSES段、TABLES段、BLOCKS段、ENTITIES段、OBJECTS段这六个段构成,其中每个段的开始标志是段的段名,而结束标志则是“ENDSEC”,每个段中内容都为ASCII码形式,故可对每个段的数据可以进行独立地读取与存储。

[0030] 本系统中主要对DXF文件中的ENTITIES(实体)段中的图元信息进行分析,需要使用的图元种类主要有Point(点)、LINE(直线)、CIRCLE(圆)、ARC(圆弧)等,根据每个图元的种类可以使用C++编写相应的图元读取程序,自动地读取相应的数据,具体的实体段数据解析流程如图2所示,首先对打开的文件获取其图形边界极小值和极大值,然后搜索ENTITIES组码进行读取,若图元为点则获取其对应坐标,若图元类型为圆弧则读取圆心坐标、半径、起始角、终止角和方向向量,若图元为直线则读取其起始点和终止点,若图元为圆则读取其圆心坐标、半径和方向向量。

[0031] 2)将对象坐标系中的加工轨迹数据转化为世界坐标系中的加工轨迹数据

[0032] 由于DXF文件中的数据结构分别定义在对象坐标系OCS和世界坐标系WCS中,三维图元(如圆和圆弧)的信息基本都定义在对象坐标系OCS中,而OCS中的数据无法直接使用,必须转换成世界坐标系WCS中的数据,如图3(b)所示,图中加粗的圆A为三维图元,该圆定义在OCS中,在进行作业任务生成时需要将圆A转换为如图3(a)所示的世界坐标系WCS中的轨迹。转换的难点在于DXF文件中未提供两坐标系间相对关系。本发明采用“任意轴原理”对坐标系进行转换,其原理是令图形软件(如AutoCAD或Micro Station)给定一单位长的向量作为OCS的Z轴,利用任意轴算法生成该坐标系相应的X轴,然后用右手法则确定Y轴,右手法则具体为:以使右手的大拇指、食指和中指三者间相互垂直,则大拇指代表X坐标,食指代表Y坐标,中指代表Z坐标,则大拇指的指向为X坐标正方向,食指的指向为Y坐标的正方向,中指的指向为Z坐标的正方向。

[0033] 具体实施步骤是检测给定对象坐标系中的Z轴是否靠近正或负的世界坐标系WCS中的X轴,如果是,由世界坐标系的Y轴与给定的Z轴(OCS中的Z轴)进行矢量积运算得到对象坐标系中的任意X轴;否则,世界坐标系的Z轴与给定的Z轴进行矢量运算得到任意X轴。其中,判别的OCS中的Z轴是否靠近WCS的Y轴的方法是判断OCS的Z轴在WCS中X轴分量 Z_x 和Y轴分量 Z_y 是否同时小于 $1/64$,若是则判别为靠近,否则为不靠近。在确定了OCS的各坐标系数数据后,即可通过WCS与OCS的相对位置关系计算出图元在WCS中的坐标。

[0034] 3)为了进行轨迹的显示与生成作业文件,对加工轨迹数据进行插补

[0035] 在获得了WCS中的轨迹数据后需要对其进行三维显示。本发明中采用图形库OpenGL显示轨迹,但是由于DXF文件中读取的基本图元信息经转换后得到的轨迹数据仍然与OpenGL进行显示的数据不相符,如DXF文件中读取的圆的数据为圆心坐标、半径、方向向量,而OpenGL中并不存在圆这一数据类型,所以需要计算圆上的多个点,然后以多条线段近似显示圆形,因此,在对轨迹进行插补产生一系列点的过程中,相邻两个点中的后一个点如何由前一个点,如图4所示,针对不同的轨迹类型采取不同的插补方式,若上一步中求得的WCS中轨迹是为直线则采用等步长法,否则采用等偏差法,等步长法所采用的变量每次以固定步长变化,等偏差法所插补出来的一系列点两两之间距离相等。等步长法和等偏差法可以由迭代公式进行解释,假设迭代公式为 $f(x_{i+1})=f(x_i)+h(\Delta x_i)$,其中 $f(x_i)$ 为前一点的值,而 $f(x_{i+1})$ 为后一点的值, Δx_i 为所选步长, $h(\Delta x_i)$ 为由步长得到的变化量,等步长法即

指 Δx_i 在每次迭代中均保持不变, 而等偏差法指 $h(\Delta x_i)$ 在每次迭代中保持不变。采用该曲线插补算法可以将 WCS 中的目标轨迹转换成能使用 OpenGL 进行显示的类型。

[0036] (2) 建立三维仿真空间, 使用虚拟现实建模语言建立待加工工件和机器人的三维实体模型并在三维仿真空间中显示, 利用所述插补后的加工轨迹数据在三维仿真空间中显示待加工工件的三维轨迹, 根据所述待加工工件的三维轨迹设置机器人的目标轨迹, 根据机器人的三维实体模型设置机器人的运动位姿。

[0037] 1) 建立三维仿真空间显示模型和目标轨迹。

[0038] 在获得加工轨迹数据后, 需要建立三维仿真环境以对轨迹显示。三维仿真空间是在计算机上生成作业文件的基础, 仿真环境的任务是首先显示仿真机器人和工件的三维实体模型以提供运动的实体和目标, 然后显示待加工工件中的三维轨迹供用户选择需要进行加工的轨迹, 即目标轨迹。

[0039] 三维场景模块是以开源软件库 OpenGL 为基础进行开发的。本发明中通过调用 OpenGL 提供的一系列三维场景建立功能在计算机上建立三维场景。在三维场景中, 使用 C++ 编写的程序模块将步骤(1)中所获得的加工轨迹数据进行显示。

[0040] 2) 根据待加工工件的三维轨迹设置机器人的目标轨迹, 根据机器人的三维实体模型设置目标位姿。

[0041] 计算机的显示器提供给用户的是一个二维平面, 而生成作业文件时使用三维仿真空间, 在三维的空间中对一条曲线的选择无法使用简单的鼠标拾取方式。在考虑上述问题之后, 本发明中提出了如图5所示的设定机器人运动轨迹的方式, 首先, 对 OpenGL 中显示的三维轨迹编写相应的转换程序转换为 Windows 平面上的二维轨迹。然后, 在程序读取鼠标的实时位置, 通过鼠标的位置距离二维轨迹的远近来给轨迹显示不同的颜色, 用户可在轨迹颜色改变后点击鼠标从而确定选择该二维轨迹。最终, 将二维轨迹转换为 OpenGL 中的三维轨迹, 即可获得机器人运动的目标轨迹。

[0042] 对于目标位姿的设定, 本平台提供了两种模式: 自动生成和用户设置。第一种方法是自动生成, 实现方法是在用户确定所采用的编程方式后即可明确姿态如何由多个曲面的约束关系生成, 然后由事先编写的程序模块根据对应的约束关系自动计算出姿态的数据。第二种方法是通过用户的设置方式, 此种方法的主要思想是设置机器人最后一个关节的坐标系的 x 轴、y 轴和 z 轴的方向来生成, 用户可以通过选择三维仿真环境中对应的轨迹来确定 x 轴和 y 轴, 此外, 用户还可以通过输入坐标系的 x 轴和 y 轴的方向来确定, 而 z 轴可以通过 x 轴和 y 轴通过常见的右手定则求得。

[0043] (3) 利用机器人的三维实体模型建立机器人运动学方程。

[0044] 本发明中采用开放式关节型机器人建模的方式建立机器人的三维实体模型, 图6 是以南京埃斯顿公司生产的六轴机器人 ER16 为例建立的机器人仿真模型, 将机器人各可以独立运动的连杆分离为 7 个部分, 在每个连杆上建立坐标系, 7 个坐标系用于确定各连杆间的位置关系, 设定连杆的旋转轴和轴向后可确定本连杆在旋转时的旋转中心轴和旋转方向, 故可获得连杆 i 相对于上一个连杆 $i-1$ 的坐标系关系 ${}^{i-1}T_i$, 故可建立机器人学中的运动学方程为 ${}^0T_6 = {}^0T_1 * {}^1T_2 * {}^2T_3 * {}^3T_4 * {}^4T_5 * {}^5T_6$, 由该方程可通过机器人学中的正逆解可以求解机器人的运动数据。通过读取机器人模型中的坐标轴数量可以确定机器人为几轴机器人, 建立不同机器人的运动学方程, 从而可以支持的机器人类型不受限制。

[0045] 此外,由于采用该种建模方式,可以动态地根据用户载入的机器人模型建立运动学方程,故可以达到支持不同种机器人同时作业的效果。

[0046] 本发明中的作业任务生成方法具有高度的灵活性,用户可以在仿真空间中通过设置机器人的位移和旋转而达到支持多种作业任务的特点,同时,可以通过设置目标轨迹的位置达到真实世界中工件轨迹可以变换的效果。由于机器人的运动学求解时是将目标轨迹转换成机器人的基坐标系下对应轨迹进行求解,而机器人和轨迹的坐标都是相对于世界坐标系而定义的,故要求进行转换,转换方法如下:机器人在世界坐标系WCS中的位移为 $Trans(x_1, y_1, z_1)$,旋转变换为 $RPY(\phi_1, \theta_1, \psi_1)$,则机器人的基坐标系在世界坐标系WCS中的齐次变换矩阵 ${}^W T_B$ 为:

$$[0047] \quad {}^W T_B = Trans(x_1, y_1, z_1) \bullet RPY(\phi_1, \theta_1, \psi_1) = \begin{pmatrix} c\phi_1 c\theta_1 & c\phi_1 s\theta_1 s\psi_1 - s\phi_1 c\psi_1 & c\phi_1 s\theta_1 c\psi_1 + s\phi_1 s\psi_1 & x_1 \\ s\phi_1 c\theta_1 & s\phi_1 s\theta_1 s\psi_1 + c\phi_1 c\psi_1 & s\phi_1 s\theta_1 c\psi_1 - c\phi_1 s\psi_1 & y_1 \\ -s\theta_1 & c\phi_1 s\psi_1 & c\theta_1 c\psi_1 & z_1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

[0048] 轨迹在世界坐标系WCS中的位置坐标为 $Trans(x_2, y_2, z_2)$,旋转变换为 $RPY(\phi_2, \theta_2, \psi_2)$,则轨迹在世界坐标系WCS中的齐次变换矩阵为 ${}^W T_P$ 为:

$$[0049] \quad {}^W T_P = Trans(x_2, y_2, z_2) \bullet RPY(\phi_2, \theta_2, \psi_2) = \begin{pmatrix} c\phi_2 c\theta_2 & c\phi_2 s\theta_2 s\psi_2 - s\phi_2 c\psi_2 & c\phi_2 s\theta_2 c\psi_2 + s\phi_2 s\psi_2 & x_2 \\ s\phi_2 c\theta_2 & s\phi_2 s\theta_2 s\psi_2 + c\phi_2 c\psi_2 & s\phi_2 s\theta_2 c\psi_2 - c\phi_2 s\psi_2 & y_2 \\ -s\theta_2 & c\phi_2 s\psi_2 & c\theta_2 c\psi_2 & z_2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

[0050] 则轨迹相对于机器人基坐标系的齐次变换矩阵 ${}^B T_P$ 为

$$[0051] \quad {}^B T_P = {}^B T_W \bullet {}^W T_P = ({}^W T_B)^{-1} \bullet {}^W T_P$$

[0052] 由以上三式可以求得目标轨迹在机器人基坐标系下的对应数据,再结合机器人运动学方程可以求解机器人运动数据。

[0053] (4)由运动数据自动生成机器人的作业文件

[0054] 从图7可以看出作业文件分为四个部分,分别为数据注释段和数据段、指令注释段和指令段。其中数据注释段和指令注释段根据用户的输入和一些条件组成。

[0055] 对于指令段的生成,根据不同的线型产生的运动指令类型和数量各不相同,首先根据轨迹的类型选择需要生成的运动指令的类型和数量,如直线对应一条MOVL指令,圆弧和圆对应三条MOVC指令,根据用户输入的每条指令的速度可以完成指令段的编辑。

[0056] 对于数据段的生成,指令段中的每条运动指令(除NOP和END外)对应于数据段中的一条指令,在数据段中根据机器人运动学方程输出机器人在各目标点对应的关节角。

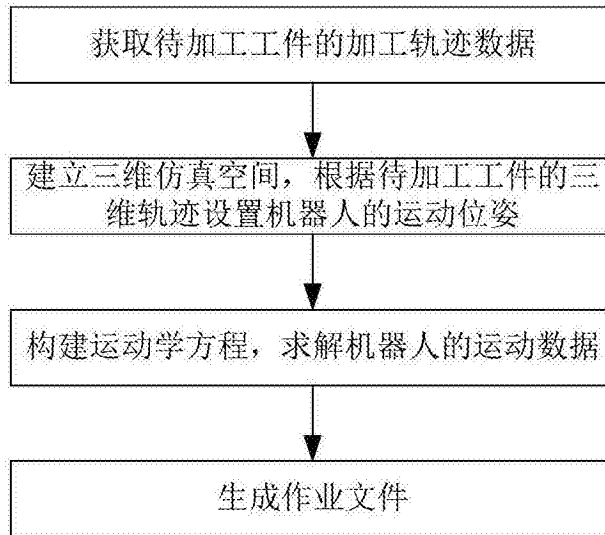


图1

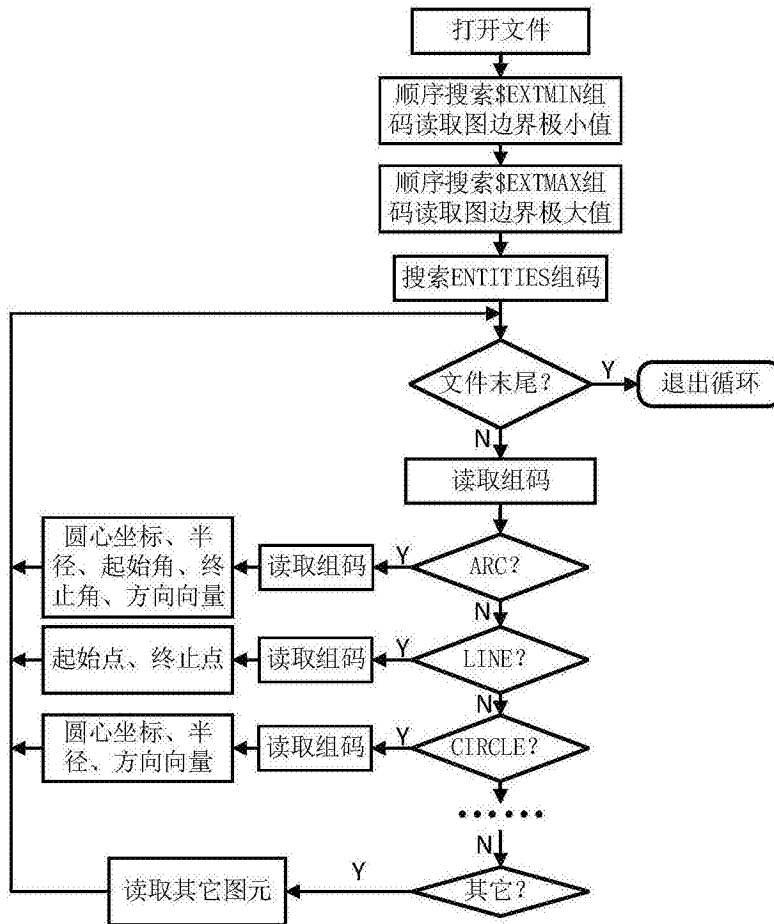


图2

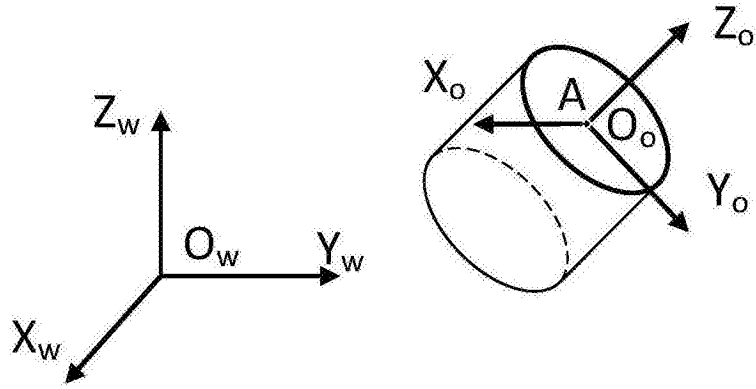


图3

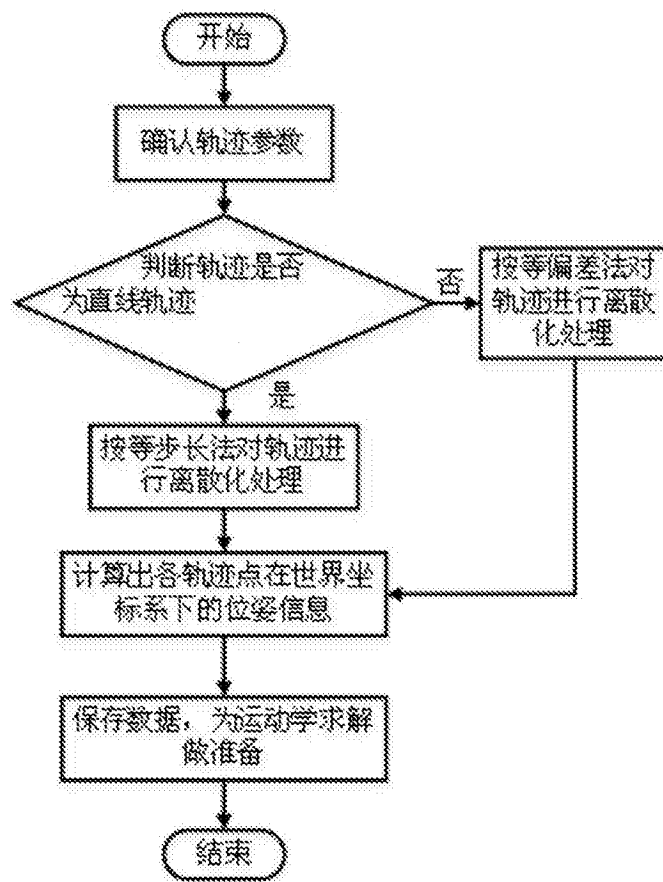


图4

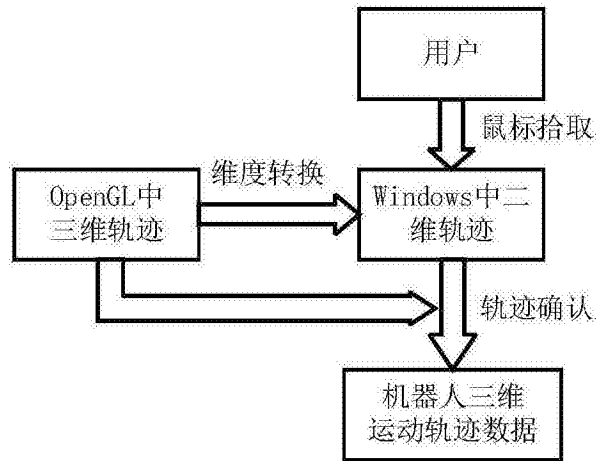


图5

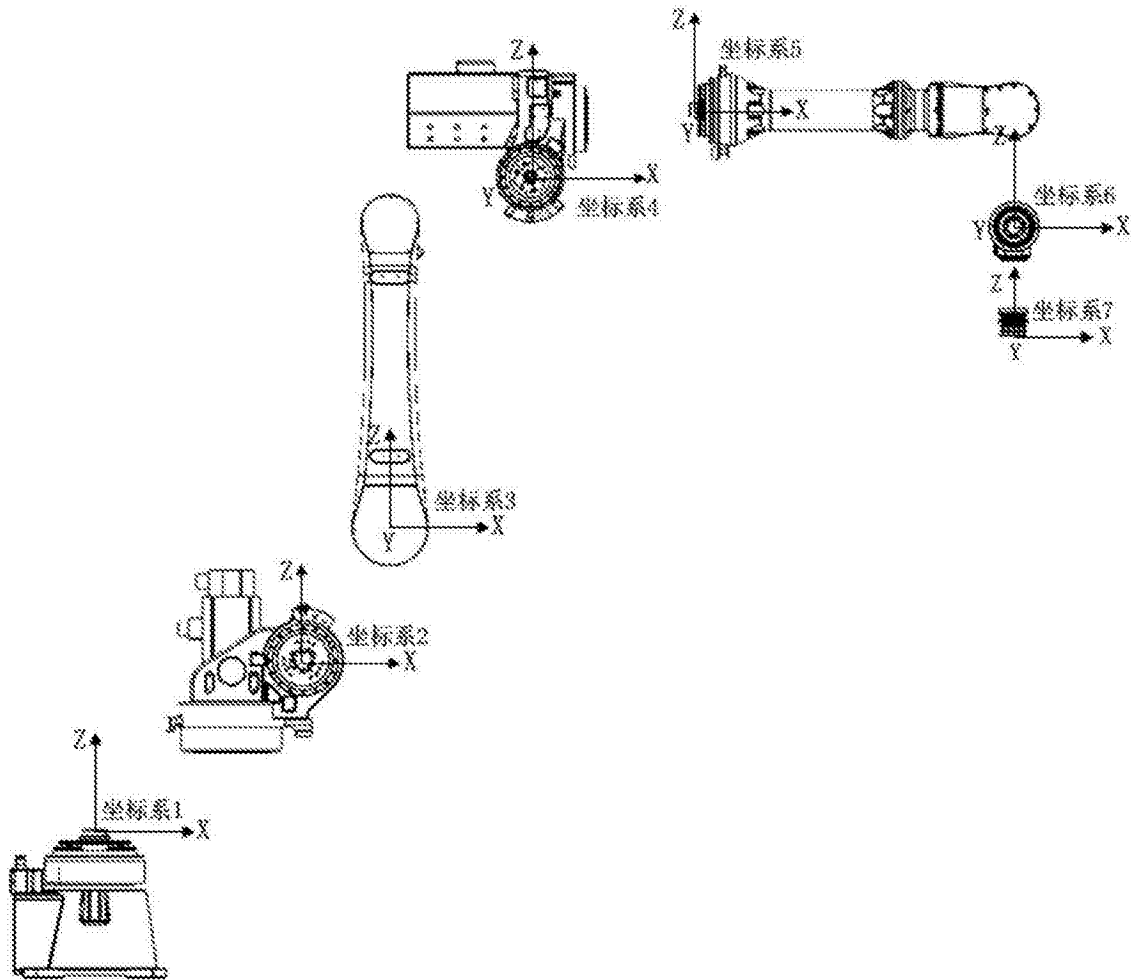


图6

<pre> /DIR /JOB //NAME ER16_job1 //POS ///NPOS 3,0,0,0,0 ///TOOL 3 ///POSTYPE PULSE ///PULSE </pre>	注释段
<pre> C0001=8.19,-1.67,-1.14,-2.69,4.27,-2.006 C0002=0.16,0.46,-0.13,-1.76,1.23,0.16 C0003=-8.31,0.44,-0.103,9.12,1.22,-8.31 C0004=8.19,-1.67,-1.14,-2.69,4.27,-2.006 </pre>	数据段
<pre> //INST ///DATE 2015/11/16/15/52 ///COMM By XJS ///ATTR SC,RW ///GROUP1 RB1 </pre>	注释段
<pre> NOP MOVJ C0001 VJ = 0.78 MOVJ C0002 VJ = 10 MOVL C0003 VJ = 10 MOVJ C0004 VJ = 0.78 END </pre>	指令段

图7