

# 一种高效的开放式关节型机器人 3D 仿真环境构建方法

甘亚辉<sup>1,2</sup>, 戴先中<sup>1,2</sup>

(1. 东南大学自动化学院, 江苏 南京 210096;

2. 复杂工程系统测量与控制教育部重点实验室, 江苏 南京 210096)

**摘要:** 提出了一种高效地构建开放式的机器人 3D 仿真环境的方法. 该方法以 Matlab 软件为平台, 充分结合 Pro/E 等 3D 实体设计软件、Matlab 机器人工具箱和 Matlab 3D 动画技术建立可视化程度高、细节显示逼真的机器人 3D 仿真研究平台. 以应用广泛的 VA1400 弧焊机器人为例, 详细介绍了机器人 3D 仿真模型的建立、仿真实体的数据导入和仿真系统的 3D 动画演示等步骤, 最终得到一个系统软件开放程度高、可视化效果逼真的机器人 3D 仿真环境. 给出了仿真系统的部分显示效果, 验证了本方法的有效性.

**关键词:** 机器人 3D 仿真; 弧焊机器人; 机器人工具箱; 3D 动画技术

中图分类号: TP273

文献标识码: A

文章编号: 1002-0446(2012)-05-0628-06

## An Efficient Method of Constructing Open 3D Simulation Environment for Articulated Robot

GAN Yahui<sup>1,2</sup>, DAI Xianzhong<sup>1,2</sup>

(1. School of Automation, Southeast University, Nanjing 210096, China;

2. Key Laboratory of Measurement and Control of Complex Systems of Engineering, Ministry of Education, Nanjing 210096, China)

**Abstract:** An efficient method for constructing an open 3D robot simulation environment is presented. Based on the platform of Matlab software, this method integrates the 3D solid design softwares such as Pro/E, Matlab Robot Toolbox and Matlab 3D visualization technology to construct a 3D robot simulation environment with good visualization and highly detailed model. Taking widely applied VA1400 arc welding robot as an example, steps of this method are presented in detail, including the construction of 3D robot model, data importation of solid model and 3D animation display of the simulation system. Thereafter, a high-level open robot simulation system with good 3D visualization is achieved. Snapshots and simulation graphs are presented, which prove the effectiveness of our method.

**Keywords:** 3D robot simulation; arc welding robot; Robot Toolbox; 3D animation

## 1 引言 (Introduction)

机器人仿真技术是机器人学研究中的一重要内容. 迄今为止, 国内外出现了许多基于 Matlab 语言编写的机器人仿真软件<sup>[1-3]</sup>. 然而目前出现的基于 Matlab 环境开发的机器人仿真软件大多采用棍图模型表示机器人, 在 3D 图形的显示方面有着明显的不足, 主要表现在: 棍图模型对机器人本体的空间构型的描述能力差, 无法直观表现机器人末端姿态和工具手姿态, 很难判断出机器人运动过程中是否与障碍物发生碰撞或者干涉. 本文的主要目的是在 Matlab 环境下增强现有的机器人等 3D 实体的显示效果, 提供一种高效地建立开放式的机器人 3D 仿

真程序的方法.

目前, 国内的绝大多数机器人仿真程序均面向工业应用<sup>[4-7]</sup>, 为教学和科研开发的具有较强开放性和交互性的机器人仿真程序几乎没有. 国外的一些面向教学应用的机器人 3D 仿真软件<sup>[2,8-9]</sup>中, 仿真程序研究的对象多为 PUMA560 机器人、Stanford 臂等较为老旧的机器人或者一些非工业用的机器人, 这与目前应用在工业领域的机器人型号相比在结构与性能上都有着较大的差异. 本文主要针对国内外研究的上述不足之处, 以目前在焊接领域广泛使用的某型工业机器人为对象, 结合现有的 3D 实体构造技术与机器人工具箱, 基于 Matlab 平台编写机器

人 3D 仿真程序，满足当前教学与科研对机器人仿真程序的需求。

## 2 仿真环境的实现步骤 (Steps of simulation environment realization)

本文选择以 Matlab 为平台，借助 Pro/E 等商业化的 3D 实体设计软件创建机器人的仿真模型，并结合现有的 Matlab 机器人工具箱为运算实现工具，利用 Matlab 的 3D 图形显示技术最终实现机器人 3D 仿真程序的设计。本文介绍的机器人仿真程序的实现大致可以分为 3 个步骤，即模型构建、模型导入和模型显示。

### 2.1 机器人本体与环境的 3D 几何造型

在机器人仿真程序的实现过程中，建立机器人本体以及仿真环境的 3D 几何造型是一项复杂而繁琐的工作。通常，人们总是希望仿真模型与真实机器人尽可能相似，这需要在模型建立过程中包含许多细节部分。但模型细节的绘制往往比较繁琐，过多的细节描绘会使建模过程花费大量的时间。因此，人们需要在绘制 3D 几何造型前，对机器人和环境物的真实形状做相应的简化处理，力求仿真模型以反映运动特征为主，而忽略那些与运动无关的具体结构细节。经过简化处理后，一个复杂的机器人本体便可以由若干个部件组成，每一个部件便可以由一些形状规则的几何体表示出来。

本文以焊接工业领域常用的 VA1400 弧焊机器人为例，具体介绍其 3D 模型建立方法。该机器人具有 7 个自由度，其照片如图 1 所示。



图 1 Motoman VA1400 机器人照片

Fig.1 Photograph of Motoman VA1400 robot

为建立细节逼真的机器人仿真模型，本文采用 Pro/E 3D 实体设计软件完成 VA1400 机器人本体的几何模型构建。该过程也可以通过其它 3D 实体设计软件（如 UG、SolidWorks、AutoCAD 等）来完成，模型构建的基本原理和实现过程基本一致。采用

Pro/E 软件建立 3D 实体几何造型的一般过程如下：

- (1) 建立模型设计的空间定位基准，如基准轴、基准面、基准坐标系等；
- (2) 建立基础的实体特征，包括拉伸、切除、旋转、扫描等；
- (3) 建立实体的工程特征，包括倒角、拔模、肋、孔等；
- (4) 添加材质和渲染处理。

以 VA1400 机器人为例，安川公司在其网站<sup>[1]</sup>提供了以 iges 格式和 step 格式存储的机器人 3D 模型，本文直接采用了该机器人 3D 模型。

机器人 3D 图形仿真所依据的基本原理是：机器人本体的几何模型由若干个部件模型拼装而成，部件间的相互位置关系由装配信息（关节的结构参数和运动参数）唯一确定，装配信息包括关节类型、关节轴取向、相邻两部件坐标架的相对位置以及关节变量。按照上述原理，首先将 Motoman VA1400 机器人的本体按照运动属性分为 8 个组成部分，即机器人 Base、S 轴、L 轴、E 轴、U 轴、R 轴、B 轴、T 轴。分解后的机器人各部分如图 2 所示。

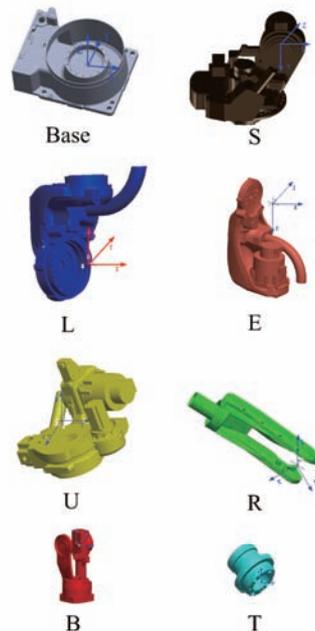


图 VA1400 机器人各组成部分

Fig.2 Components of Motoman VA1400 robot

### 2.2 仿真实体的模型导入方法

完成机器人本体以及仿真环境的几何造型设计后，需要将仿真实体的 3D 模型导入到 Matlab 环境中。由于 Matlab 软件本身不能识别 3D 图形文件，因此研究人员必须自己设计并编写 Matlab 程序，完成仿真实体 3D 模型到 Matlab 环境的导入。目前，仿真实体的 3D 模型可以采用多种格式存储，如 iges

格式、step 格式、slp 格式、wrl 格式等。

本文采用 slp 文件格式保存 3D 实体的仿真模型，并设计相应的文件导入程序。采用 slp 文件格式的优点是：slp 文件采用 ASCII 码存储，文件组成形式简单，文件内容的可读性强。典型的 slp 文件的格式如图 3 所示：第 1 行是零件的名称，接下来是多行的平面和顶点数据，然后是颜色说明以及符合这些颜色的面和点，直到下一个颜色指定出现。

依据上述 slp 文件的格式分析结果，本文设计

```

solid BLOCK
color 1.000 1.000 1.000
facet
normal 0.000000e+00 0.000000e+00 -1.000000e+00
normal 0.000000e+00 0.000000e+00 -1.000000e+00
normal 0.000000e+00 0.000000e+00 -1.000000e+00
outer loop
vertex 5.000000e-01 -5.000000e-01 -5.000000e-01
vertex -5.000000e-01 -5.000000e-01 -5.000000e-01
vertex -5.000000e-01 5.000000e-01 -5.000000e-01
endloop
endfacet

```

图 3 典型的 slp 文件格式  
Fig.3 Typical slp file format

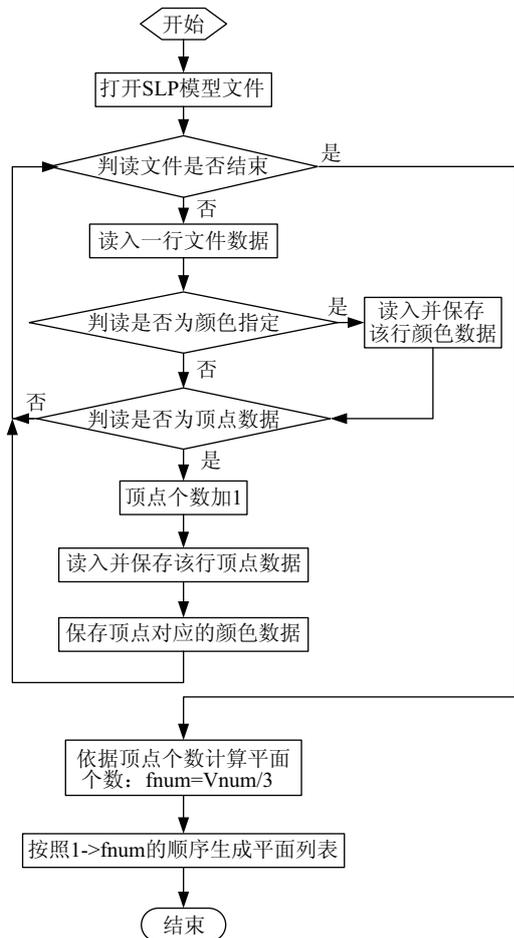


图 4 slp 文件导入流程  
Fig.4 Importation flowchart of slp file

了将 slp 文件导入到 Matlab 中的程序，其工作流程如图 4 所示。按照图 4 的流程编写的 Matlab 函数为

```
function [fout, vout, cout] = SLP_Importor(filename)
```

该函数以要读入的文件名 filename.slp 为入口参数，读取以 ASCII 码形式存储的 \*.slp 文件，返回 3D 实体的面列表、顶点列表和颜色信息作为 Matlab 图形显示命令的参数，其中返回值 fout 是面列表，vout 是顶点列表，cout 是颜色信息。

### 2.3 仿真模型的 3D 显示与动画演示

完成 slp 文件的导入后，接下来是将上述导入的 3D 实体的顶点数据、面数据和颜色数据借助 Matlab 的 3D 图形显示命令显示出来。Matlab 的 3D 可视化功能中提供了 patch 函数，可用于 3D 图形的显示，本文采用了如下格式的 patch 函数调用形式用于显示指定的 3D 实体模型：

```
handle = patch('PropertyName', propertyvalue, ...)
```

该函数可以用来为显示对象的显示属性赋值，显示对象的显示属性 PropertyName 包括显示对象的平面属性“faces”、顶点信息“vertices”等，handle 是用来操作显示对象的句柄。以 VA1400 机器人的 S 轴为例，借助 patch 函数将其在 Matlab 中以 3D 图形显示出来的代码如下所示：

```

%%%%%%%%%%
% 生成空白显示图
f=figure('name','RobotSimulator', 'WindowStyle','normal', ...
'color',[ 0.5 0.5 0.5], 'units', 'normalized', 'position',[1,1,8,8]);
% 读入要现实的实体零部件 [平面、顶点、颜色]
[fout2, vout2, cout2] = SLP_Importor('soliddata/S.slp');
% 为每一个实体指定操作句柄，赋值显示属性
p2 = patch('faces', fout2, 'vertices', vout2);
% 设置显示属性
set(p2, 'facec', 'flat', 'FaceVertexCData', cout2, 'EdgeColor','none');
% 设置默认灯光
light;
% 设置坐标轴的显示单位比率
daspect([1 1 1]);
% 设置观测视角属性
view(3);
% 设置图形属性坐标轴名称、标题和范围
xlabel('X'),ylabel('Y'),zlabel('Z');
title(['Axis S of Motoman VA1400 ']);
range = 500; % or can be changed
axis([-range-200 range-200 -range range -500 range]);
grid;
% 将读入的图形直接显示出来
drawnow;

```

执行上述代码可以在 Matlab 环境中得到如图 5 所示的显示效果。

图 5 显示的机器人的 S 轴 3D 模型的形状是正确的，但摆放的位置与实际机器人有差别。这是因

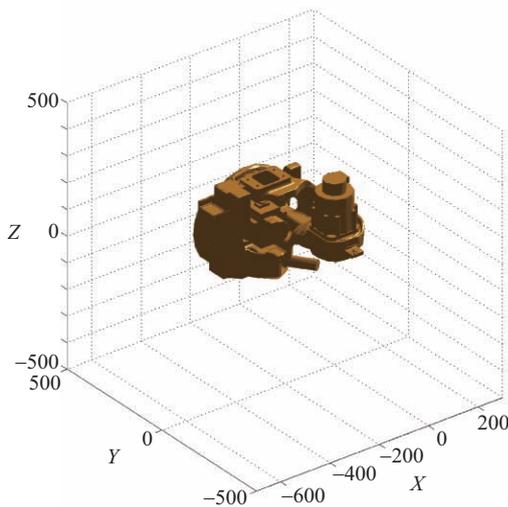


图 5 VA1400 机器人的 S 轴在 Matlab 中的 3D 显示效果  
Fig.5 3D view for Axis S of VA1400 robot in Matlab

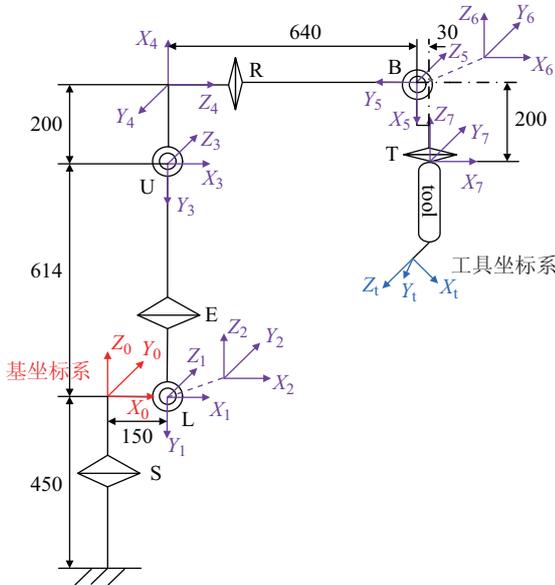


图 6 VA1400 机器人各部件坐标系  
Fig.6 Frames for each component of VA1400 robot

表 1 VA1400 机器人 DH 参数表  
Tab.1 DH parameters of VA1400 robot

Axis	$\theta_i$	$\alpha_i$	$a_i/\text{mm}$	$d_i/\text{mm}$	$\theta_i$ 范围
S	$\theta_1$	$-90^\circ$	150	0	$-170^\circ \sim 170^\circ$
L	$\theta_2$	$90^\circ$	0	0	$-70^\circ \sim 148^\circ$
E	$\theta_3$	$-90^\circ$	0	614	$-90^\circ \sim 90^\circ$
U	$\theta_4$	$-90^\circ$	200	0	$-175^\circ \sim 150^\circ$
R	$\theta_5$	$-90^\circ$	0	640	$-150^\circ \sim 150^\circ$
B	$\theta_6$	$90^\circ$	30	0	$-45^\circ \sim 180^\circ$
T	$\theta_7$	0	0	-200	$-200^\circ \sim 200^\circ$

为在绘制机器人各部件的 3D 造型时都选择了独立的设计基准。要将机器人各部件按照正确的顺序显

示出来，必须确定这些设计基准间的空间位姿关系。本文按照 Denavit-Hartenberg 方法为每一个机器人部件建立基准坐标系，并采用 DH 参数描述坐标系间的位姿关系。采用 DH 表示法的 VA1400 机器人各部件基准坐标系的建立结果如图 6 所示，坐标系间的 DH 参数如表 1 所示，其中  $\theta_i$  为关节变量， $\alpha_i$  为关节扭角， $a_i$  为连杆长度， $d_i$  为两连杆距离。

由 DH 参数表示的机器人相邻两部件间的位姿变换矩阵可由式 (1) 表示，式中  $c(\cdot)$  表示  $\cos(\cdot)$ ， $s(\cdot)$  表示  $\sin(\cdot)$ ， ${}^{i-1}T_i$  表示坐标系  $i$  到  $i-1$  的变换关系。

$${}^{i-1}T_i = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & \alpha_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -d_i s\alpha_{i-1} \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & d_i c\alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

按照 DH 方法建立机器人各部件坐标系后，为了按照正确的装配信息显示各部件，必须重新计算机器人各部件 3D 模型中的顶点位置。对于串联的垂直关节型机器人来说，每个关节的运动具有向后承继性，即机器人各关节的运动带动后面的部件运动，但后面关节的运动不会影响前面部件的位姿。借助式 (1) 表示的齐次变换矩阵，机器人各部件的顶点位置按照每个关节的运动会发生如下变化：

$$\begin{aligned} {}^0T_1 &= {}^0T_1(\theta_1) \\ {}^0T_2 &= {}^0T_1(\theta_1) \cdot {}^1T_2(\theta_2) \\ {}^0T_3 &= {}^0T_1(\theta_1) \cdot {}^1T_2(\theta_2) \cdot {}^2T_3(\theta_3) \\ {}^0T_4 &= {}^0T_1(\theta_1) \cdot {}^1T_2(\theta_2) \cdot {}^2T_3(\theta_3) \cdot {}^3T_4(\theta_4) \\ {}^0T_5 &= {}^0T_1(\theta_1) \cdot {}^1T_2(\theta_2) \cdot {}^2T_3(\theta_3) \cdot {}^3T_4(\theta_4) \cdot {}^4T_5(\theta_5) \\ {}^0T_6 &= {}^0T_1(\theta_1) \cdot {}^1T_2(\theta_2) \cdot {}^2T_3(\theta_3) \cdot {}^3T_4(\theta_4) \cdot {}^4T_5(\theta_5) \cdot {}^5T_6(\theta_6) \\ {}^0T_7 &= {}^0T_1(\theta_1) \cdot {}^1T_2(\theta_2) \cdot {}^2T_3(\theta_3) \cdot {}^3T_4(\theta_4) \cdot {}^4T_5(\theta_5) \cdot {}^5T_6(\theta_6) \cdot {}^6T_7(\theta_7) \end{aligned}$$

文 [1] 介绍了一种采用 Matlab 语言编写的机器人工具箱，提供了丰富的机器人学相关运算和变换函数。本文充分借助此机器人工具箱的运算功能，完成一种新的机器人 3D 仿真程序的编写。在 Matlab 机器人工具箱中提供了 SerialLink 类型的变量用于表示机器人，借助 SerialLink 型变量的 link 类型的成员变量 links 可实现对机器人各关节的访问。link 类型的变量又提供了  $4 \times 4$  的矩阵型的成员变量 A 可用于完成式 (1) 中的运算。相关程序的部分代码如下所示：

```

% 设置机器人各关节的DH参数
theta = [0 0 0 -pi/2 pi -pi/2 0];
% theta d a alpha
L(1) = Link([ 0 0 150 -pi/2]);
L(2) = Link([ 0 0 0 pi/2]);
.....
L(7) = Link([ 0 -200 0 0]);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 连接为VA1400机器人
VA1400 = SerialLink(L, 'name', 'VA1400', 'manufacturer', 'YASKAWA
clear L;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 依据机器人关节DH参数重新设置各个零部件的基准坐标系, 按照J
nv1 = vout1;
set(p1, 'Vertices', vout1(1:3,:));
nv2 = VA1400.links(1).A(theta(1))*vout2;
set(p2, 'Vertices', nv2(1:3,:));
nv3 = VA1400.links(1).A(theta(1))*VA1400.links(2).A(theta(2))*vout3;
set(p3, 'Vertices', nv3(1:3,:));
.....

```

执行该段代码后得到如图 7 所示的显示效果。

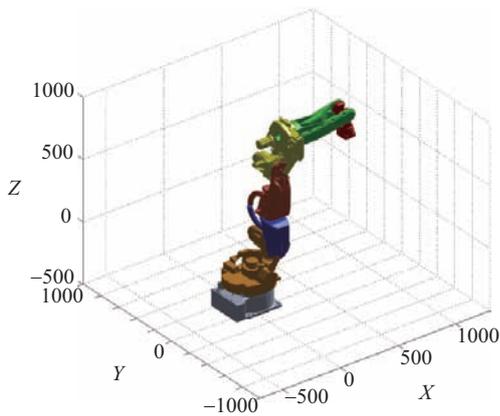


图 7 Matlab 中的 VA1400 机器人 3D 仿真图  
Fig.7 3D simulation graph of VA1400 robot in Matlab

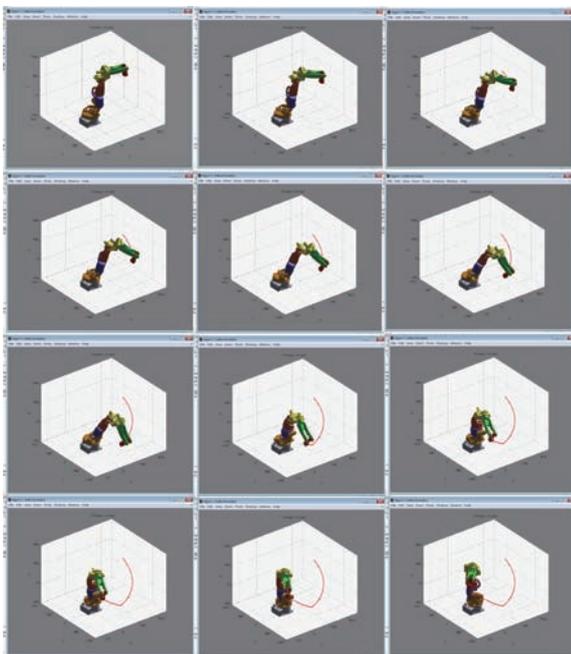


图 8 VA1400 机器人仿真运动截图  
Fig.8 Snapshot for motion of VA1400 robot

按照上述步骤, 已经可以实现机器人 3D 仿真模型在 Matlab 中的静态显示, 进一步还需要实现上述机器人 3D 模型的动画演示. Matlab 中的动画显示方式主要有电影方式和物件方式两种: 电影方式以帧的形式存储要播放的图像, 显示速度快, 画面比较精细, 但无法根据用户的要求交互性地改变显示结果; 物件方式通过不断绘制新的图像替代原有图像实现动画演示, 演示过程的交互能力较强, 但显示计算和图形刷新需要耗费较多的时间. 由于机器人仿真过程需要较高的交互性, 本文最终选定以物件方式实现动画演示. 具体来说, 就是通过不断地修改机器人的关节变量, 并按照当前关节变量值重新计算机器人部件的顶点位置, 依据最新的顶点位置不断刷新显示图形而获得动画效果. 示例性的某段 VA1400 机器人的运动过程截图如图 8 所示.

多机器人协作系统的研究是当前机器人领域的热点, 目前的机器人仿真系统针对多机器人协作过程的仿真功能一般都比较弱. 本文介绍的方法也可以用来创建包含多个机器人的仿真系统. 通过引入仿真机器人的基座位姿信息, 就可以实现多机器人的显示. 对于多机器人仿真系统而言, 需要明确每个机器人的基坐标系到世界坐标系的关系, 通过左乘机器人基坐标系到世界坐标系的变换矩阵就可以将不同的机器人在仿真环境中正确的显示出来. 计算过程如式 (2) 所示.

$${}^{\text{world}}T_n = {}^{\text{world}}T_{\text{base}} \cdot {}^0T_1(\theta_1) \cdot {}^1T_2(\theta_2) \cdots {}^{n-1}T_n(\theta_n) \tag{2}$$

本文以 Motoman VA1400 机器人和 HP20 机器人为例, 通过上述方法实现双机器人系统的协调运动和同步动画显示, 在 Matlab 仿真程序中的显示效果如图 9 所示.

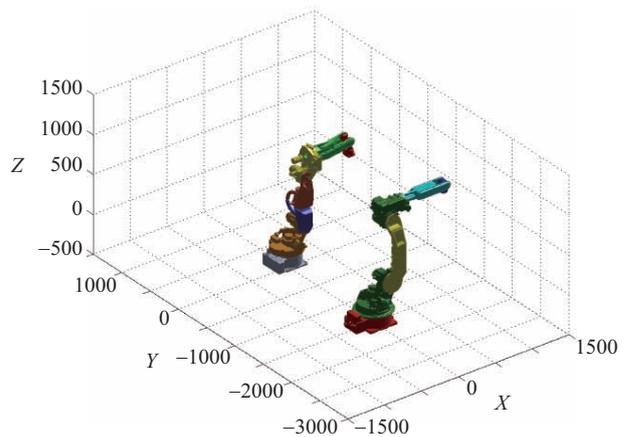


图 9 双机器人仿真 3D 图形  
Fig.9 3D graph of dual robots simulation

### 3 本方法优点 (Advantages of this method)

通过以上介绍,本文实现了以 Matlab 为平台的机器人 3D 仿真程序.同第 1 节中提到的其它的机器人仿真程序相比,该方法具有如下优点:

(1) 仿真程序基于 Matlab 实现,可充分利用 Matlab 平台强大的数据处理能力,同时保证了程序在仿真过程中具有较强的交互性;

(2) 充分利用了现有的成熟技术实现仿真程序,包括 3D 实体设计软件、Matlab 机器人工具箱、Matlab 3D 图形显示技术等,开发效率高,仿真程序的开放性好;

(3) 充分利用了机器人生产公司提供的 3D 模型,避免了仿真模型建立的繁琐过程,且仿真模型的精度高、细节逼真;

(4) 借助 Matlab 机器人工具箱实现底层运算,避免了机器人学中基本运算过程的实现,可以保证使用对象关注更高层次的规划和运算,充分面向教学与科研型应用;

(5) 借助商用化的 3D 实体建模软件建立仿真模型,设计了标准化的 slp 格式 3D 模型导入函数,保证了仿真程序具有一个开放的机器人类型库。

### 4 结论 (Conclusion)

本文提出了一种基于 Matlab 平台的机器人 3D 仿真程序构建方法,该方法充分结合 Pro/E 等 3D 实体设计软件、Matlab 机器人工具箱、Matlab 3D 图形显示技术,增强了现有的 Matlab 机器人仿真程序的 3D 图形显示与动画演示功能,特别是对于普通高校的机器人教学工作具有如下意义:

(1) 高度细节化的机器人 3D 模型使得机器人学的问题更具直观性;

(2) 提供了大量机器人学中的基本方法和运算求解函数;

(3) 开放性的软件结构和强大的数据显示处理能力为机器人智能规划方法和离线编程系统的设计提供了验证平台。

由于 Matlab 语言本身的执行速度较慢,代码的执行效率较低.本文以 Matlab 为平台构建机器人 3D 仿真软件,并非为开发商用化的机器人软件,而是以满足机器人的教学和演示为目的.与商用化的机器人离线编程或仿真软件相比,本文的研究工作和取得的结果可以说是微不足道的.但是,该方法充分结合并应用了现有的多种成熟的先进技术,包括计算理论、计算机图形学、机器人学的诸多进展,解决自己的研究领域中出现的问题,因而是一项非常有意义的工作.有关本文程序的代码可联系作者

获取,希望本文的工作能够为促进机器人的教学和普及略尽绵力,能够为其他研究者的机器人相关研究工作提供一些启示。

### 参考文献 (References)

- [1] Corke P I. MATLAB toolboxes: Robotics and vision for students and teachers[J]. Robotics & Automation Magazine, 2007, 14(4): 16-17.
- [2] Kucuk S, Bingul Z. An off-line robot simulation toolbox[J]. Computer Applications in Engineering Education, 2010, 18(1): 41-52.
- [3] 王健美,付成龙,黄元林,等.基于 Matlab 的双足机器人动力学仿真及仿生控制平台[J].系统仿真学报,2011,23(5): 977-983.  
Wang J M, Fu C L, Huang Y L, et al. Dynamics simulation and bionic control platform for biped robot with Matlab[J]. Journal of System Simulation, 2011, 23(5): 977-983.
- [4] 孙增圻,严隽薇,范锡恩,等. PUMA 机器人的动态仿真[J]. 机器人, 1987, 1(5): 1-5.  
Sun Z Q, Yan J W, Fan X E, et al. Dynamic simulation of PUMA manipulators[J]. Robot, 1987, 1(5): 1-5.
- [5] 丁汉,陈家新,熊有伦. 机器人最佳轨迹规划和图形仿真的研究[J]. 机器人, 1989, 3(2): 8-14.  
Ding H, Chen J X, Xiong Y L. A study on optimal trajectory planning and graphic simulation of robot manipulators[J]. Robot, 1989, 3(2): 8-14.
- [6] 熊震宇,王冬祥. OpenGL 在 MOTOMAN 弧焊机器人动态仿真中的应用[J]. 上海交通大学学报, 2008, 42(S1): 87-89.  
Xiong Z Y, Wang D X. Application of OpenGL in 3D animated simulation system of MOTOMAN arc-welding robot[J]. Journal of Shanghai Jiaotong University, 2008, 42(S1): 87-89.
- [7] 柴婷婷,丁言露. 机器人仿真及其自动化研究进展[J]. 制造业自动化, 2011, 33(5): 87-89.  
Chai T T, Ding Y L. The development of the robotic simulation and automation[J]. Manufacturing Automation, 2011, 33(5): 87-89.
- [8] Qassem M A, Abuhadrous I, Elaydi H. Modeling and simulation of 5 DOF educational robot arm[C]//2nd IEEE International Conference on Advanced Computer Control. Piscataway, NJ, USA: IEEE, 2010: 569-574.
- [9] Toz M, Kucuk S. Dynamics simulation toolbox for industrial robot manipulators[J]. Computer Applications in Engineering Education, 2010, 18(2): 319-330.
- [10] ABB Group. RobotStudio 简介 [OL]. [2011-12-27]. <http://www.abb.com.cn/product/seitp327/5859a6f2147f09a2482570e40033a505.aspx>.
- [11] Yaskawa Motoman Robotics. VA1400 简介 [OL]. [2011-12-27]. <http://www.motoman.eu/products/robots-nx-xrc-and-dx/product-view/>.

### 作者简介:

甘亚辉 (1985-), 男, 博士生. 研究领域: 智能化焊接机器人, 多机器人协作控制, 无夹具焊接过程控制。

戴先中 (1954-), 男, 博士, 教授. 研究领域: 复杂控制理论, 机器人控制, 电力系统控制, 测量及信号处理。