**《机器人学》第一次作业**

1. 计算题，提交**纸版**，提交时间：**2017.3.14 12:55**，地点：课堂。
2. Matlab编程题，提交**电子版**【包括Word文档写出答案和结果（文件名“**机器人学第一次作业\_姓名\_学号.docx**”）和Matlab “.m”文件】，提交时间：**2017.3.21 12:55**之前。发送email to：胡航 (kemain@sjtu.edu.cn)，CC to： [y.ding@sjtu.edu.cn](mailto:y.ding@sjtu.edu.cn); mexiong@sjtu.edu.cn
3. **计算题**
4. 假设给定三个坐标系*o*1*x*1*y*1*z*1、*o*2*x*2*y*2*z*2和*o*3*x*3*y*3*z*3，同时假设



计算。

1. 求出齐次变换矩阵, 它表示坐标系{*B*}（初始时与{*A*}重合）相对于固定坐标系{*A*}作了以下变换：
2. 绕*y*A轴转30度；(b) 再绕*x*A轴转-90度；(c) 再绕*z*A轴转90度；(d) 最后作移动(2, 6, 9)T。
3. 求出齐次变换矩阵, 它表示相对于固定坐标系{*A*}和运动坐标系{*B*}（初始时与{*A*}重合）作了以下变换：
4. 移动(2, 4, 9)T；(b) 绕*x*B轴转-90度；(c) 绕*z*A轴转90度；(d) 绕*y*B轴转90度。
5. 求下面齐次变换矩阵



的逆变换矩阵。

1. 矩阵



表示齐次变换矩阵，求其中第一列元素。请进一步求出其旋转矩阵所对应的转动轴和转角（初始时与固定坐标系重合）。

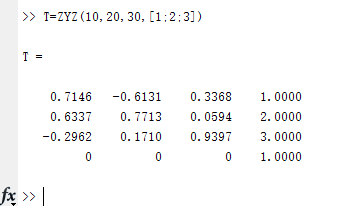
1. **Matlab编程题**
2. 编写Matlab函数，要求函数输入ZYZ欧拉角()和位置向量，输出齐次变换矩阵，

并用两组数据测试结果：

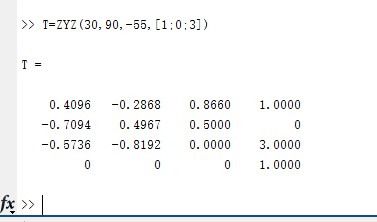
1. ，，，；
2. ，，，。

**解：M文件命名为：ZYZ.m**

1. **的测试结果为：**

****

1. **的测试结果为：**

****

**源代码为：**

function [ T ] = ZYZ( Z1,Y1,Z2,o )

Z1=Z1/180\*pi;

Y1=Y1/180\*pi;

Z2=Z2/180\*pi;

o=o.';

Rz1=[cos(Z1),-sin(Z1),0;sin(Z1),cos(Z1),0;0,0,1];

Ry1=[cos(Y1),0,sin(Y1);0,1,0;-sin(Y1),0,cos(Y1)];

Rz2=[cos(Z2),-sin(Z2),0;sin(Z2),cos(Z2),0;0,0,1];

R=Rz1\*Ry1\*Rz2;

T=[R,o.';zeros(1,3),1];

end

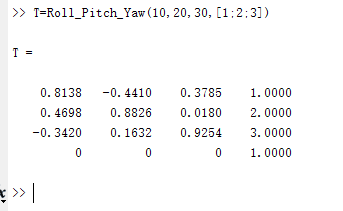
1. 编写Matlab函数，要求函数输入Roll–Pitch–Yaw欧拉角()和位置向量，输出齐次变换矩阵，

并用两组数据测试结果：

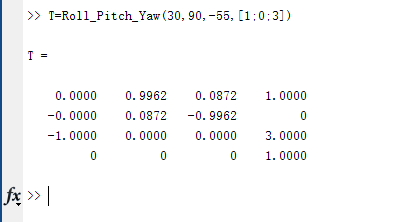
1. ，，，；
2. ，，，。

**解：M文件命名为：Roll\_Pitch\_Yaw.m**

1. **的测试结果为：**

****

1. **的测试结果为：**

****

**源代码为：**

function [ T ] = Roll\_Pitch\_Yaw( Roll,Pitch,Yaw,o )

Roll=Roll/180\*pi;

Pitch=Pitch/180\*pi;

Yaw=Yaw/180\*pi;

o=o.';

Ryaw=[cos(Yaw),-sin(Yaw),0;sin(Yaw),cos(Yaw),0;0,0,1];

Rpitch=[cos(Pitch),0,sin(Pitch);0,1,0;-sin(Pitch),0,cos(Pitch)];

Rroll=[1,0,0;0,cos(Roll),-sin(Roll);0,sin(Roll),cos(Roll)];

R=Ryaw\*Rpitch\*Rroll;

T=[R,o.';zeros(1,3),1];

End

1. 编写三个Matlab函数，要求函数输入旋转 矩阵，分别输出ZYZ欧拉角、Roll–Pitch–Yaw欧拉角以及可一次实现该旋转的单位向量***r***和转角*θ*，并验证程序结果的正确性。

**解：M文件分别命名为：Q3\_ZYZ.m，Q3\_Roll\_Pitch\_Yaw.m，Q3\_r\_theta.m，Q3\_test.m(该程序用于验证上述程序的正确性)**

**Q3\_ZYZ.m源代码为：**

function [ Z1,Y1,Z2 ] = Q3\_ZYZ( R )

%ZYZ欧拉角

while abs(det(R)-1)>10^(-6)

R=input('请重新输入旋转矩阵R=');

end

Z1=atan2(R(2,3),R(1,3));

Y1=atan2((R(1,3)^2+R(2,3)^2)^0.5,R(3,3));

Z2=atan2(R(3,2),-R(3,1));

End

**%函数结果得到方法实例：[Z1,Y1,Z2]=Q3\_ZYZ(R)**

**Q3\_Roll\_Pitch\_Yaw.m的源代码为：**

function [ Roll,Pitch,Yaw ] = Q3\_Roll\_Pitch\_Yaw( R )

R=input('请输入3\*3旋转矩阵R=');

while abs(det(R)-1)>10^(-6)

R=input('请重新输入旋转矩阵R=');

end

Roll=atan2(R(3,2),R(3,3));

Pitch=atan2(-R(3,1),(R(3,2)^2+R(3,3)^2)^0.5);

Yaw=atan2(R(2,1),R(1,1));

end

**%函数结果得到方法实例：[Roll,Pitch,Yaw]=Q3\_Roll\_Pitch\_Yaw(R)**

**Q3\_r\_theta.m源代码为：**

function [ r,theta ] = Q3\_r\_theta( R )

%绕r旋转theta角度

while abs(det(R)-1)>10^(-6)

R=input('请重新输入旋转矩阵R=');

end

theta=acos((R(1,1)+R(2,2)+R(3,3)-1)/2);

r=(1/(2\*sin(theta)))\*[R(3,2)-R(2,3);R(1,3)-R(3,1);R(2,1)-R(1,2)];

if theta==0

r=[1;0;0];

end

end

**%函数结果得到方法实例：[r,theta]=Q3\_r\_theta(R)**

**%值得注意的是，本函数若旋转角度计算为0的话，默认旋转单位向量为[1;0;0]**

**Q3\_test.m(该程序用于验证上述程序的正确性)源代码为：**

clear all

clc

R=input('请输入3\*3旋转矩阵R=');

[Z1,Y1,Z2]=Q3\_ZYZ(R);

[Roll,Pitch,Yaw]=Q3\_Roll\_Pitch\_Yaw(R);

[r,theta]=Q3\_r\_theta(R);

%验证Q3\_ZYZ的正确性

Rz1=[cos(Z1),-sin(Z1),0;sin(Z1),cos(Z1),0;0,0,1];

Ry1=[cos(Y1),0,sin(Y1);0,1,0;-sin(Y1),0,cos(Y1)];

Rz2=[cos(Z2),-sin(Z2),0;sin(Z2),cos(Z2),0;0,0,1];

R1=Rz1\*Ry1\*Rz2

error1=norm(R1-R);

D=['ZYZ欧拉角的误差为：',num2str(error1)];

disp(D);

%验证Q3\_Roll\_Pitch\_Yaw的正确性

Ryaw=[cos(Yaw),-sin(Yaw),0;sin(Yaw),cos(Yaw),0;0,0,1];

Rpitch=[cos(Pitch),0,sin(Pitch);0,1,0;-sin(Pitch),0,cos(Pitch)];

Rroll=[1,0,0;0,cos(Roll),-sin(Roll);0,sin(Roll),cos(Roll)];

R2=Ryaw\*Rpitch\*Rroll

error2=norm(R2-R);

E=['Roll-Pitch-Yaw欧拉角的误差为：',num2str(error2)];

disp(E);

%验证Q3\_r\_theta的正确性

if r(1)==0&r(2)==0

R3=[cos(theta),-sin(theta),0;sin(theta),cos(theta),0;0,0,1]

else

alpha=asin(r(2)/(r(1)^2+r(2)^2)^0.5);

beta=acos(r(3));

X1=[cos(alpha),-sin(alpha),0;sin(alpha),cos(alpha),0;0,0,1];

X2=[cos(beta),0,sin(beta);0,1,0;-sin(beta),0,cos(beta)];

X3=[cos(theta),-sin(theta),0;sin(theta),cos(theta),0;0,0,1];

X4=[cos(-beta),0,sin(-beta);0,1,0;-sin(-beta),0,cos(-beta)];

X5=[cos(-alpha),-sin(-alpha),0;sin(-alpha),cos(-alpha),0;0,0,1];

R3=X1\*X2\*X3\*X4\*X5

end

error3=norm(R3-R,2);

F=['第三种方法的误差为：',num2str(error3)];

disp(F);

**%程序结果：**

**if error1<10^(-6)&error2<10^(-6)&error3<10^(-6)**

**disp('程序正确');**

**else**

**disp('程序计算错误');**

**end**