

上海工程技术大学

实验报告



课 程	《机器人虚拟仿真实验》
实 验 名	基于 Matlab 的机器人运动仿真与轨迹规划 基于机器人虚拟仿真系统的机器人应用仿真
学 院	机械与汽车工程学院
专 业	机器人工程
学 号	011921220
姓 名	吴杰
指 导 教 师	朱姿娜、武欣
完成时间	2023.12.11

目 录

实验一 基于 Matlab 的机器人运动仿真与轨迹规划·····	3
实验二 基于机器人虚拟仿真系统的机器人应用仿真·····	10
实验心得 ·····	13

实验 1：基于 Matlab 的机器人运动仿真与轨迹规划

1. 实验目的

- 1) 熟悉 MATLAB 编程环境；
- 2) 学习 MATLAB 基本知识；
- 3) 掌握 MATLAB 的 Robotics 工具箱的使用。

2. 实验工具

- 1) MATLAB 软件；
- 2) Robotics Toolbox 工具箱。

3. 实验原理

- 1) 通过 MATLAB 软件实现复杂的数值运算；
- 2) 通过添加 Robotics Toolbox 实现机器人仿真；
- 3) 通过 MATLAB 软件的 Robotics Toolbox 工具箱建立机器人，并用其中的运动学函数，计算机器人复杂的正、逆运动学解。
- 4) 用 Robotics Toolbox 工具箱的轨迹规划函数，对机器人的搬运轨迹进行仿真与规划。

4. 实验步骤

- 1) 启动 MATLAB 软件并熟悉软件操作页面的菜单栏、工具栏、文件编辑窗口、命令窗口、历史命令窗口、工作空间等；
- 2) 安装 Robotics Toolbox 工具箱，并激活工具箱；
- 3) 激活命令窗口，进行简单运算；
- 4) 练习创建一维、二维数组；
- 5) 练习用 Robotics 工具箱进行简单的坐标变换及建立机器人模型；
- 6) 用 Robotics 工具箱里的“fkine”和“ikine”函数对所建立的机器人模型进行正逆运动学计算，并求出雅克比矩阵；
- 7) 用 Robotics 工具箱里的“jtraj”和“ctrjaj”函数对所建立的五自由度机器人模型进行轨迹规划；
- 8) 将机器人模型、轨迹规划曲线用 plot 函数显示出来。

5. 实验内容

- 1) 通过 MATLAB 软件计算下列方程的值；

$$\left(\sqrt{2e^{x+0.5}+1}\right), \quad y = \frac{2\sin(0.3\pi)}{1+\sqrt{5}}, \quad y = \frac{2\cos(0.3\pi)}{1+\sqrt{5}}$$

2) 创建数组 a，并将其第 2 行的元素均替换成 1，给出程序执行的程序和结果截图贴入实验报告；

$$a = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$$

3) 利用 Robotics toolbox 工具箱，求解：坐标系 {B} 的初始位姿与参考坐标系 {A} 相同，坐标系{B}相对于{A}的 z 轴旋转 30°，再沿{A}的 x 轴移动 12，沿{A}的 y 轴旋转 45 度，求该坐标变换的齐次矩阵。

4) 建立五自由度机器人模型的几何参数如下表所示，命名为“五自由度机器人”，并用 plot 显示该机器人（将程序和机器人 plot 图形截图贴入实验报告）。

连杆 i	a_{i-1}	α_{i-1}	d_i	θ_i
1	0	0°	0	θ_1
2	0	-90°	0	θ_2
3	1	0°	0	θ_3
4	1	0°	0	θ_4
5	0	90°	0	θ_5

5) 对该机器人进行正、逆运动学求解。

(1) 正运动学计算时，已知 $\theta_1=45^\circ, \theta_2=-60^\circ, \theta_3=30^\circ, \theta_4=45^\circ, \theta_5=-60^\circ$ 。

(2) 逆运动学计算时，已知末端姿态为 (1) 中正运动学的求解结果。

6) 对该机器人进行关节空间、直角坐标空间轨迹规划。

(1) 用“jtraj”函数，已知初始和终止的关节角度，利用五次多项式来规划轨迹；初始角度 $\theta_1=0^\circ, \theta_2=0^\circ, \theta_3=0^\circ, \theta_4=0^\circ, \theta_5=0^\circ$ ；终止角度 $\theta_1=\pi/4, \theta_2=-\pi/3, \theta_3=\pi/5, \theta_4=\pi/2, \theta_5=-\pi/4$ 。

(2) 用“ctrjaj”函数，已知初始和终止的末端关节位姿，利用匀加速、匀减速运动来规划轨迹；

```
p1 = [ -0.7071    0    0.7071    1.4142;
        0        -1     0         0;
        0.7071    0    0.7071    1.9142;
        0         0     0         1]
```

```
p2 = [ 0.9640   -0.2639   -0.0332    0.9331;  
       0.0979    0.2361    0.9668    1.9331;  
      -0.2473   -0.9352    0.2534    0.8618;  
       0         0         0         1]
```

(3) 显示机器人模型、“ctrj”函数轨迹规划的末端关节轨迹、“jtraj”函数轨迹规划的各关节的角度，角速度，角加速度。

6. 思考题

1) 简述如何运用 Robotics Toolbox 建立机器人模型。

1、安装 Robotics Toolbox：首先，确保已经在 MATLAB 环境中安装了 Robotics Toolbox。可以从官方网站或其他可靠来源下载并安装该工具箱。2、导入 Robotics Toolbox：在 MATLAB 中，使用 addpath 命令将 Robotics Toolbox 的路径添加到 MATLAB 的搜索路径中。3、创建机器人对象：使用 Robotics Toolbox 提供的函数，创建一个机器人对象。该对象包含机器人的几何信息、关节参数和运动学模型等。4、设置机器人参数：根据机器人的实际参数，使用机器人对象的属性和方法设置机器人的 DH 参数、关节限制、连杆长度、质量分布等参数。

2) 逆运动学计算结果是否与正运动学计算时给定的关节变量值一致？若不一致，试分析是什么原因造成的？

在求解过程中可能得到一个近似解，而不是唯一解。在进行数值计算时，存在舍入误差和数值近似等问题。机器人在某些位置和姿态上可能出现奇异点，即无法进行逆运动学求解的点。机器人可能有关节限制条件，如关节角度范围、连杆长度限制等。如果在逆运动学求解过程中违反了这些限制条件，那么逆运动学计算结果可能与正运动学给定的关节变量值不一致。

3) 简述用 robotics 工具箱求解机器人运动学的优势。

统一框架、多样性支持、灵活性、效率高、结合仿真环境

4) 对比两种轨迹规划函数的特点和区别。

jtraj 函数：

特点：jtraj 函数是 MATLAB Robotics Toolbox 中的一个函数，用于在关节空间内进行插值轨迹规划。它可以生成平滑的关节空间轨迹，通过指定起始关节角、目标关节角、时间间隔等参数，自动生成关节空间插补轨迹。

区别：jtraj 函数通常用于机器人关节空间的规划，能够生成符合机器人关节运动范围和速度限制的平滑轨迹。它适用于需要在关节空间内进行运动规划的情况，例如机器人的关节空间运动控制和路径规划。

ctrj 函数：

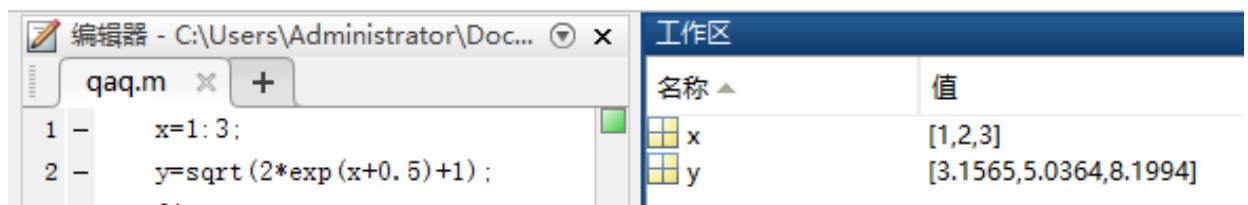
特点：ctrj 函数同样是 MATLAB Robotics Toolbox 中的一个函数，用于在笛卡尔空间内进行插值轨迹规划。它可以生成平滑的笛卡尔空间轨迹，通过指定起始位姿、目标位姿、时间间隔等参数，自动生成笛卡尔空间插补轨迹。

区别：ctrj 函数主要用于机器人末端执行器的轨迹规划，能够生成符合机器人末端执行器运动范围和速度限制的平滑轨迹。它适用于需要在笛卡尔空间内进行运动规划的情况，例如机器人末端执行器的路径规划和轨迹控制。

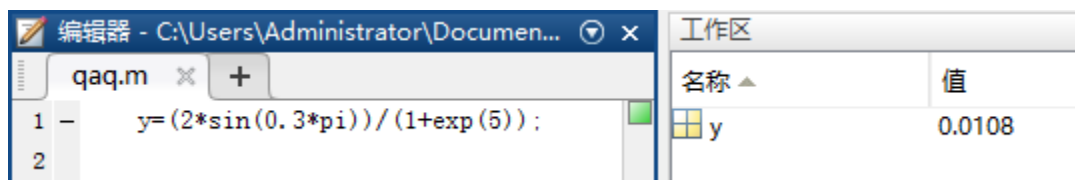
实验内容题目解答：

1.

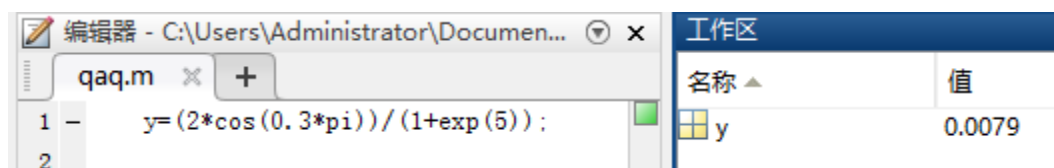
(1)



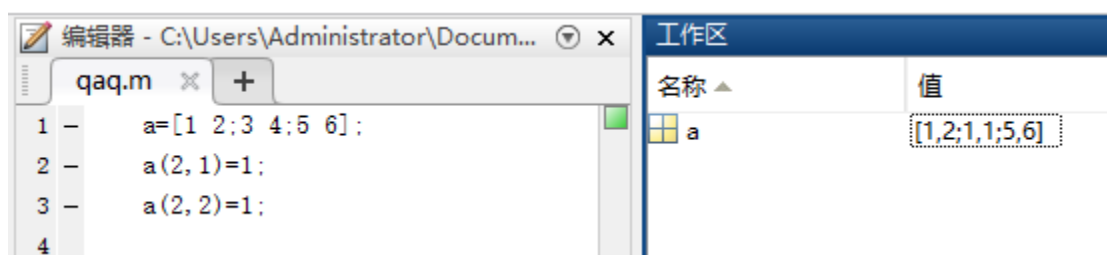
(2)



(3)



2.



3.

```

编辑器 - C:\Users\Administrator\Desktop\qqq\Untitled6.m
Q6.m  Q5.m  Untitled6.m  +
1 - A=eye(4);
2 - R=trjoty(pi/4)*transl(12,0,0)*trotz(pi/6);
3 - B=R*A
4
命令窗口
>> Untitled6

B =

    0.6124   -0.3536    0.7071    8.4853
    0.5000    0.8660         0         0
   -0.6124    0.3536    0.7071   -8.4853
         0         0         0    1.0000

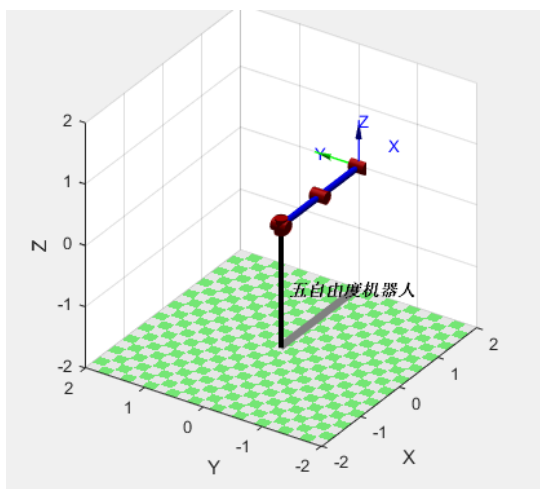
```

4.

```

编辑器 - C:\Users\Administrator\Desktop\qqq\Q5.m
Q5.m  Q6.m  +
1 - clear;
2 - clc;
3 - L1 = Link('d', 0, 'a', 0, 'alpha', 0); %Link 类函数
4 - L2 = Link('d', 0, 'a', 0, 'alpha', -pi/2);
5 - L3 = Link('d', 0, 'a', 1, 'alpha', 0);
6 - L4 = Link('d', 0, 'a', 1, 'alpha', 0);
7 - L5 = Link('d', 0, 'a', 0, 'alpha', pi/2);
8
9 - b=isrevolute(L1); %Link 类函数
10 - robot=SerialLink([L1,L2,L3,L4,L5]); %SerialLink 类函数
11 - robot.name='五自由度机器人'; %SerialLink 属性值
12 - robot.manuf='five'; %SerialLink 属性值
13 - robot.display(); %Link 类函数
14 - theta=[0 0 0 0 0];
15
16 - theta1=[pi/4,-pi/3,pi/6,pi/4,-pi/3,];
17 - robot.plot(theta);
18 - p1=robot.fkine(theta1);
19

```



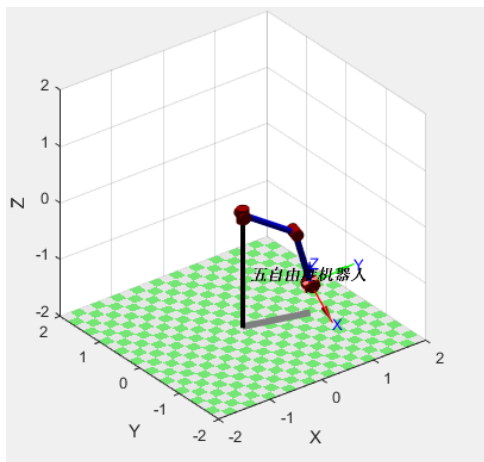
5.

(1)

```

编辑器 - C:\Users\Administrator\Desktop\qq\Q5.m
Q5.m Q5.m trchain.m +
1 clear;
2 clc;
3 L1 = Link('d', 0, 'a', 0, 'alpha', pi/2); %Link 类函数
4 L2 = Link('d', 0, 'a', 0.5, 'alpha', 0, 'offset', pi/2);
5 L3 = Link('d', 0, 'a', 0, 'alpha', pi/2, 'offset', pi/4);
6 L4 = Link('d', 1, 'a', 0, 'alpha', -pi/2);
7 L5 = Link('d', 0, 'a', 0, 'alpha', pi/2);
8
9 b=revolute(L1); %Link 类函数
10 robot=SerialLink([L1,L2,L3,L4,L5]); %SerialLink 类函数
11 robot.name='五自由度机器人'; %SerialLink 属性值
12 robot.manuf='five'; %SerialLink 属性值
13 robot.display(); %Link 类函数
14 theta=[0 0 0 0 0];
15 robot.plot(theta); %SerialLink 类函数
16
17 theta1=[pi/4,-pi/3,pi/6,pi/4,-pi/3,];
18
19 p1=robot.fkine(theta1)

```



p1 =

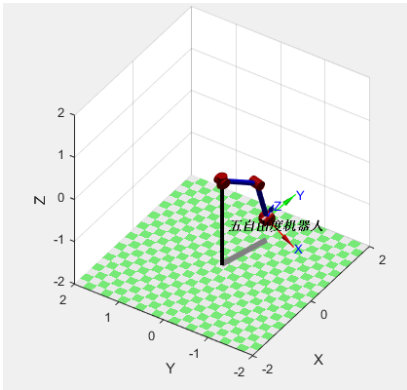
-0.0991	0.9330	-0.3459	1.0865
-0.8700	-0.2500	-0.4250	-0.2911
-0.4830	0.2588	0.8365	-1.4659
0	0	0	1.0000

(2)

```

编辑器 - Q5.m
Q5.m Q6.m +
1 clear;
2 clc;
3 L1 = Link('d', 0, 'a', 0, 'alpha', 0); %Link 类函数
4 L2 = Link('d', 0, 'a', 0, 'alpha', -pi/2);
5 L3 = Link('d', 0, 'a', 1, 'alpha', 0);
6 L4 = Link('d', 0, 'a', 1, 'alpha', 0);
7 L5 = Link('d', 0, 'a', 0, 'alpha', pi/2);
8 L6 = Link('d', 0, 'a', 0, 'alpha', pi/2);
9
10 b=revolute(L1); %Link 类函数
11 robot=SerialLink([L1,L2,L3,L4,L5,L6]); %SerialLink 类函数
12 robot.name='五自由度机器人'; %SerialLink 属性值
13 robot.manuf='five'; %SerialLink 属性值
14 robot.display(); %Link 类函数
15
16 theta=[0 0 0 pi/2 0];
17 theta1=[pi/4,-pi/3,pi/6,pi/4,0,-pi/3];
18
19 robot.plot(theta1);
20
21 p1=robot.fkine(theta1); %正运动学
22 p2=robot.ikine(p1); %逆运动学
23
24 j0=robot.jacob0(p2); %雅可比矩阵

```

j0 =

0.4774	0.4774	-0.7082	-0.4984	0	0
0.3008	0.3008	1.1239	0.7909	0	0
0.0000	0.0000	-0.5643	0.3550	0	0
0.0000	0.0000	0.8461	0.8461	0.8461	0.2940
-0.0000	-0.0000	0.5331	0.5331	0.5331	-0.4666
1.0000	1.0000	0.0000	0.0000	0.0000	0.8342

p2 =

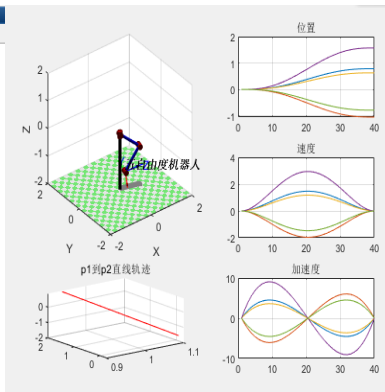
2.6373 2.6373 0.4045 1.5293 -1.3496 -1.8765

6.

```

1 clear;
2 clc;
3 L1 = Link('d', 0, 'a', 0, 'alpha', 0); %Link 类函数
4 L2 = Link('d', 0, 'a', 0, 'alpha', -pi/2);
5 L3 = Link('d', 0, 'a', 1, 'alpha', 0);
6 L4 = Link('d', 0, 'a', 1, 'alpha', 0);
7 L5 = Link('d', 0, 'a', 0, 'alpha', pi/2);
8
9 robot=SerialLink([L1,L2,L3,L4,L5]); %SerialLink 类函数
10 robot.name='五自由度机器人'; %SerialLink 属性值
11 robot.aamuf='欧'; %SerialLink 属性值
12
13 init_ang=[0 0 0 0 0];
14 targ_ang=[pi/4,-pi/3,pi/5,pi/2,-pi/4];
15 step=40;
16 [q,qd,qdd] = jtraj(init_ang, targ_ang, step);
17 subplot(3,2,[1,3]);
18 robot.plot(q);
19
20 subplot(3,2,2);
21 z=1:5;

```



实验 2：基于机器人虚拟仿真系统的机器人应用仿真

1. 实验目的

- 1) 熟悉机器人虚拟仿真软件环境；
- 2) 学习机器人虚拟仿真软件建模方法；
- 3) 掌握建立简单机器人应用场景的虚拟仿真模型。

2. 实验工具

- 1) 计算机；
- 2) ROBOGUIDE 软件。

3. 实验原理

- 1) 通过机器人虚拟仿真软件 ROBOGUIDE 建立机器人的三维数模；
- 2) 通过机器人虚拟仿真软件 ROBOGUIDE 搭建机器人应用场景；
- 3) 通过机器人虚拟仿真软件 ROBOGUIDE 编程实现机器人运动模拟。

4. 实验步骤

- 1) 启动 ROBOGUIDE 软件；
- 2) 新建工作单元、创建指定型号机器人、设置选择参数、完成创建；
- 3) 工作单元建成后，生成三维数模，进入工作空间，熟悉软件操作界面的菜单栏、工具栏、显示窗口等；
- 4) 熟悉机器人的关节坐标系、直角坐标系（世界坐标系、手动坐标系、用户坐标系）、工具坐标系；
- 5) 打开示教器，在示教器里切换不同坐标系进行机器人运动演示；
- 6) 在机器人末端安装夹具、搭建机器人工作环境（工作台、料框、工件等）；
- 7) 通过编程实现机器人在指定场景下的运动模拟。
- 8) 生成动画并保存。

5. 思考题

- 1) 简述如何运用 ROBOGUIDE 软件建立工作单元。

创建新项目：在 ROBOGUIDE 中，首先创建一个新的项目，并设置工作单元的参数，例如工作区域的大小、机器人类型等。

添加设备：在项目中添加机器人、传感器、工件夹具和其他设备，以便建立完整的工作单元模型。

建立布局：利用 ROBOGUIDE 的建模工具，在虚拟环境中建立工作单元的布局，包括机器人工作区域、工作台、传送带等。

2) 如果想要调试机器人末端相对于工作台的位置，在哪个坐标系下调节比较合适？

世界坐标系

3) 简述如何用 ROBOGUIDE 软件建立多个阵列工件。

创建基本工件模型：使用 CAD 软件或 ROBOGUIDE 提供的建模工具创建一个基本的工件模型，可以是一个单一的零件或组件。

复制工件：选择已创建的工件模型，在 ROBOGUIDE 中使用复制功能进行复制。可以指定复制的数量、间距和方向等参数。

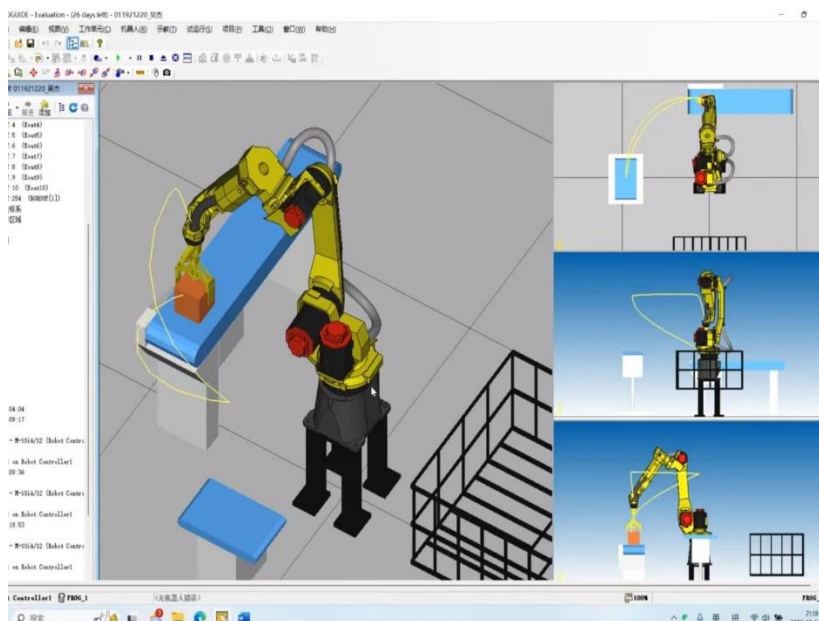
排列工件：将复制的工件进行排列，可以通过拖拽、旋转或输入坐标来调整工件的位置。确保工件之间的间距和位置符合要求。

创建阵列：选择排列好的工件，使用 ROBOGUIDE 提供的阵列功能进行阵列操作。可以选择直线阵列、矩形阵列或自定义阵列等类型。

调整阵列参数：根据需要，调整阵列的参数，如数量、间距、偏移量、排列方向等。通过预览功能，查看阵列效果的变化。

优化布局：对于大规模阵列工件，可能需要进一步优化布局以提高工作效率和路径规划。使用 ROBOGUIDE 的路径规划和碰撞检测功能进行布局优化。

验证仿真：使用 ROBOGUIDE 的仿真功能验证阵列工件的位置、机器人的路径和姿态等是否符合要求，检查是否存在碰撞或其他问题。



实验心得

. 基于机器人虚拟仿真系统进行机器人应用仿真的实验可以提供很多有益的心得和经验。以下是一些可能的实验心得：

. 提前规划和准备：在进行机器人应用仿真实验之前，进行充分的规划和准备非常重要。这包括理解实验的目标和要求，选择适当的仿真软件和模型，以及准备好所需的环境设置和输入参数。

. 熟悉仿真工具：熟悉所使用的机器人虚拟仿真系统和相关工具是至关重要的。了解仿真软件的功能、界面和操作方法，能够有效地利用各种工具来创建和模拟机器人应用场景。

. 精确建模和参数设置：在进行机器人仿真时，精确建模和准确设置参数是关键步骤。确保机器人模型和环境模型的准确性，以及传感器、执行器和控制器等参数的正确设置，可以获得更真实和可靠的仿真结果。

. 多样化场景测试：尝试在不同的场景下进行仿真实验，包括不同的环境、任务和工作条件。这可以帮助验证机器人在各种情况下的性能和适应性，并且有助于发现潜在的问题和改进方案。

. 数据分析和评估：对仿真结果进行数据分析和评估是非常重要的。通过观察和比较不同参数、算法或策略的仿真结果，可以获得对机器人应用性能的深入理解，并为实际应用提供参考和优化建议。

. 文档记录和分享：及时记录实验过程、结果和心得体会，并与他人分享经验，可以帮助巩固学习成果，促进交流和合作，以及为今后的研究和开发工作提供有价值的参考。

总之，基于机器人虚拟仿真系统进行机器人应用仿真的实验是一种有效的方法，可以帮助理解和验证机器人系统的性能和功能。通过充分准备、合理设置参数、多样化场景测试和数据分析，可以获得有意义的实验心得和经验，并为实际应用提供指导和改进方向。