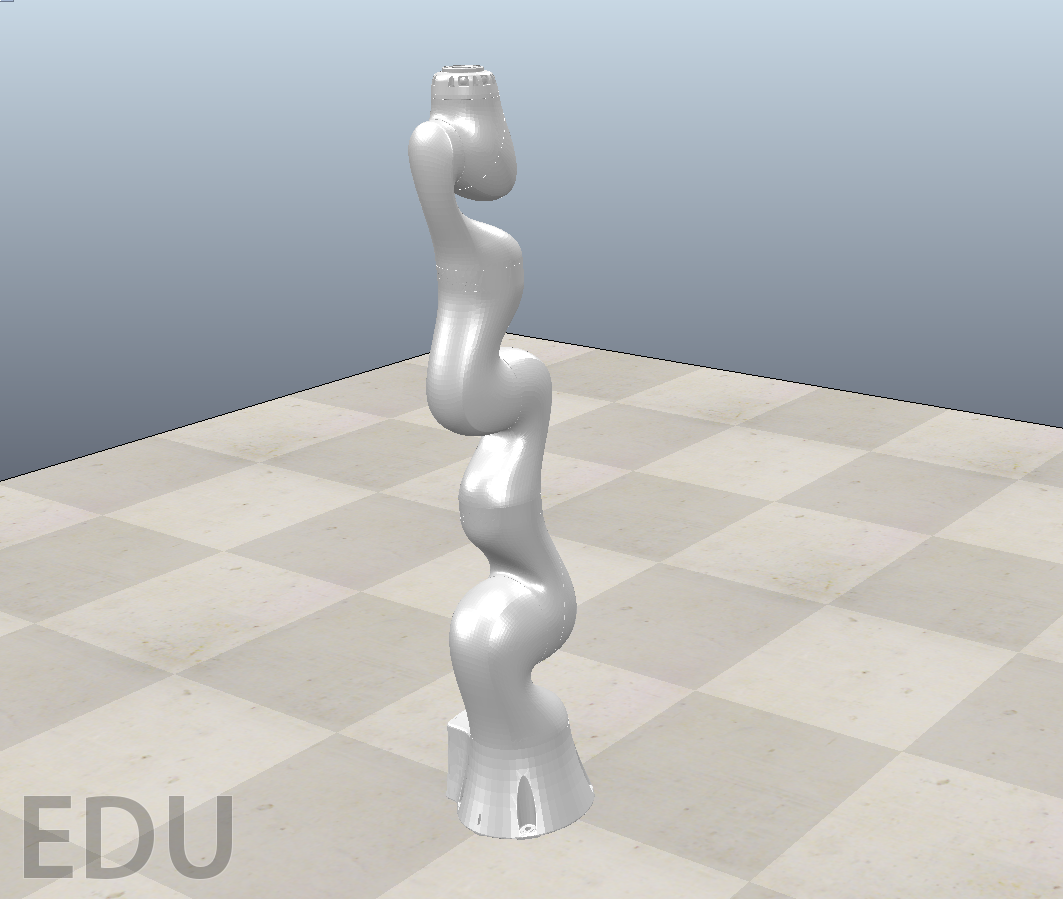
考虑如下的七自由度KUKA LBR iiwa机器人：



MATLAB自带LBR iiwa的运动学、动力学文件*iiwa14.urdf*

调用其动力学模型、运动学模型、前向运动学Jacobian矩阵的Api可点击链接：

<https://www.mathworks.com/help/robotics/ug/robot-dynamics.html>

<https://github.com/yanseim/pd_plus_gravity_control>

以上链接包括了关节空间“PD+重力补偿”控制算法示例，用公式表达为：



以上控制方法的其他细节可参看课件。

**作业要求：**

1. 设计**任务空间控制算法**，驱动机械臂末端从[0.5;-0.2;0.5]移至[0.4;-0.3;0.4]。描述所设计算法的优势与不足，证明稳定性。提示：基于LaSalle不变集定理。

注：控制过程中无须保持末端方向不变。文件task\_space\_control.ttt中机械臂末端的位置即[0.5;-0.2;0.5]，将iiwa\_link\_ee\_kuka\_visual的位置作为末端位置。

1. 设计**无需精确动力学参数的轨迹追踪控制算法**，驱动机械臂末端从[0.5;0;0.5]追踪如下时变轨迹。描述设计思路，证明稳定性。提示：基于Barbalat引理。

可参考论文Neural Net Robot Controller with Guaranteed Tracking Performance

提示：

1. 可以按课上讲的方法，在关节空间中定义滑模向量。将闭环方程中部分未知的项用一个RBF网络估计：把f看成x的函数（x是包含的向量，这么看会引入误差），RBF网络输入为x，输出为，激活层为径向基函数，其中为中心，为方差。这两个参数需要人为给定。上述过程可以表示为

代码表示为（数字是随便给的，需要自己调参）

1. c = ones(dimension\_of\_x,1) \* linspace(-1.5,1.5,node\_num);
2. b=5;
4. phi = zeros(node\_num,1);
5. **for** j = 1:node\_num
6. phi(j) = exp(-norm(x-c(:,j))^2/(b^2));
7. end
9. f = W\_hat' \* phi
10. 也可以直接在笛卡尔空间中定义滑模向量，最后算得的末端力通过投影回关节空间，其余步骤同1。也可以用多个RBF网络分别估计不同的项。方法很多，能完成任务即可。
11. 设计**基于零空间的控制算法**，在（第4、第6关节）冗余关节受到外部扰动的情况下（），保持末端位置不变化（效果参看之前链接）。描述设计思路，证明稳定性。提示：参看课件中的冗余关节运动学分析思路。
12. 提交报告、程序代码、仿真结果（包括但不限于：末端轨迹运动图、任务空间位置误差图、关节扭矩图等）、证明过程，以及其他相关内容。