基于零空间方法（NUB）的全身控制（WBC）的简单实现

# 一，问题背景

机械臂通常是6自由度，因为这是执行三维空间任务的最少自由度。虽然6自由度机械臂能满足多数任务，但是机器人的机械臂应该像人的手臂一样更加灵活。当机器人的连杆在工作空间中需要在避开障碍物的同时跟踪末端执行器的给定轨迹时，就需要比在无障碍工作空间中更多的自由度。另外，在足式机器人，比如两足机器人上，通常有用于行走任务的腿部关节和用于执行用户任务的上肢关节，而这些关节数量一般远大于6。

对于上述多任务高自由度机器人，存在以下两个特点：

1. 通常同时存在多个控制任务，如末端轨迹跟随和连杆避障，如下肢任务与上肢任务。
2. 在一些关节奇异位置，可能无法同时满足所有控制任务，必须舍弃一些。因此，引入任务优先级的概念：

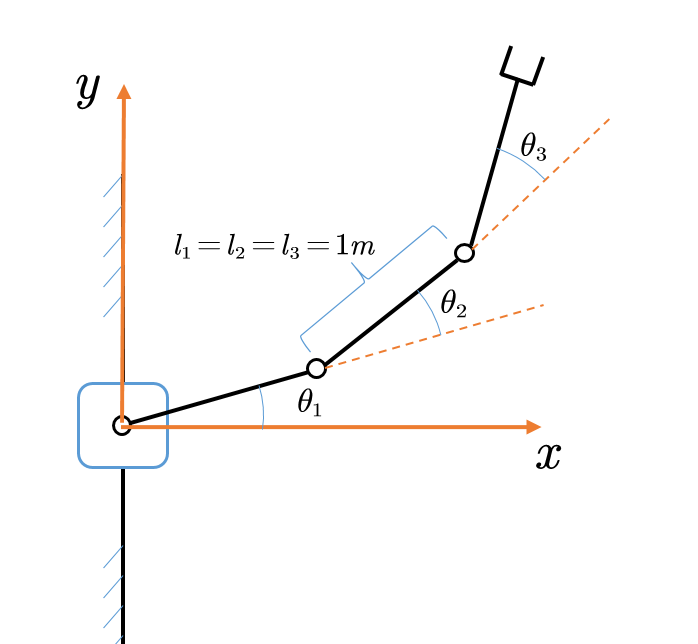
我们有时会发现末端执行器位置要比方向更重要。例如，在焊接、激光切割等工作中，必须尽可能保证末端执行器位置精度。

我们有时也会发现末端执行器方向要比位置更重要。比如，机械臂末端是一台天文望远镜，当观测非常远的天文现象时，机械臂位置误差并不重要，但方向上稍有误差，就可能无法对准观测任务。

因此，一个综合地机器人控制任务，可以依据不同的意义分解成不同的子任务，并根据重要程度划分优先级。在必要时，低优先级任务不应该影响高优先级任务的执行。

# 二，实际的例子

动力学建模不是本文重点，因此本文控制系统采用速度控制，不考虑动力学问题。



**图1**

如图1，在一个直线轨道上有一个滑块，沿着y轴自由运动，滑块上有平面3R机械臂，各连杆长度均为1m，定义刚性连接到滑块上，且与世界坐标系方向相同的坐标系为滑块坐标系。设计3R机械臂控制系统，满足以下要求：

* 控制系统输入：滑块在轨道上的位置，机械臂各关节位置。
* 控制系统输出：机械臂各个关节的速度。
* 控制任务1：让机械臂末端位置保持在世界坐标系下点。
* 控制任务2：让机械臂末端朝向x轴。

## 2.1 无优先级的控制方案

根据上述控制任务，在滑块坐标系下，机械臂末端的期望坐标为：



在滑块坐标系下，已知各关节坐标求机械臂末端坐标的过程称为运动学正解：



其中，，依次类推。由于，下面的计算中将省略。

控制器本身的设计不是本文重点，采用最简单的P控制：



将运动学正解微分，可得：



其中, 是雅克比矩阵：



设符号为雅克比矩阵的伪逆，可以计算控制器的输出，即关节速度为：



注意这里伪逆的性质：若有逆，则是的逆；若没有逆，可满足的最小二乘解。

在webots2019b下搭建仿真环境，采用matlab搭建控制系统，效果视频如下：

视频1，机械臂末端满足控制任务1与控制任务2.

视频2，机械臂期望位置超出机械臂可达工作空间，系统崩溃。

核心代码：

% controller

xd = [2 -yh 0]';

x = [c1+c12+c123 s1+s12+s123 theta1+theta2+theta3]';

x\_dot = kp\*(xd-x);

J = [-s1-s12-s123 -s12-s123 -s123; c1+c12+c123 c12+c123 c123; 1 1 1];

pinvJ = pinv(J);

theta\_dot = pinvJ\*x\_dot;

## 2.1 位置优先级高于方向优先级

在冗余机器人上通常，公式（1.1）有更广义的表达形式：



其中是满足行数的任意向量。对于控制任务来说，为零向量矩阵，相当于一个“过滤器”，把控制无关的机器人自由度过滤掉。即，无论是什么，都不会对控制产生任何影响。这样，我们就可以通过来控制更低优先级的控制任务，同时不影响高优先级的任务。

根据控制任务1和控制任务2，将机械臂位姿坐标分解成两个变量：



用和分别表示控制任务1与控制任务2的期望坐标：



同样采用简单的P控制器：



在之前的工作中，已经求出和的雅克比矩阵分别为：



位置优先级高于方向优先级时，将雅克比矩阵代入公式1.2，可得：



在webots2019b下搭建仿真环境，采用matlab搭建控制系统，效果视频如下：

视频，位置优先级高于方向优先级

从视频中可以看出，当滑块位置离其初始位置很远时，机械臂无法同时满足末端的位置任务和姿态任务，因此在一定程度上放弃姿态任务的控制，尽可能保证位置控制任务。

核心代码

% controller

r1d = [2 -yh]';

r2d = [0];

r1 = [c1+c12+c123 s1+s12+s123]';

r2 = [theta1+theta2+theta3];

r1\_dot = r1\_kp\*(r1d-r1);

r2\_dot = r2\_kp\*(r2d-r2);

J1 = [-s1-s12-s123 -s12-s123 -s123; c1+c12+c123 c12+c123 c123];

J2 = [1 1 1];

pinvJ1 = pinv(J1);

pinvJ2 = pinv(J2);

I = eye(3)

theta\_dot = pinvJ1\*r1\_dot + (I - pinvJ1\*J1)\*pinvJ2\*r2\_dot;

## 2.2 位置优先级低于方向优先级

将公式1.3稍作改变，交换位置任务和姿态任务的优先级：



在webots2019b下搭建仿真环境，采用matlab搭建控制系统，效果视频如下：

视频，位置优先级低于方向优先级

从视频中可以看出，当滑块位置离其初始位置很远时，机械臂无法同时满足末端的位置任务和姿态任务，因此在一定程度上放弃姿态任务的控制，尽可能保证位置控制任务。

核心代码

% controller

r1d = [2 -yh]';

r2d = [0];

r1 = [c1+c12+c123 s1+s12+s123]';

r2 = [theta1+theta2+theta3];

r1\_dot = r1\_kp\*(r1d-r1);

r2\_dot = r2\_kp\*(r2d-r2);

J1 = [-s1-s12-s123 -s12-s123 -s123; c1+c12+c123 c12+c123 c123];

J2 = [1 1 1];

pinvJ1 = pinv(J1);

pinvJ2 = pinv(J2);

I = eye(3)

theta\_dot = pinvJ2\*r2\_dot + (I – pinvJ2\*J2)\*pinvJ1\*r1\_dot;