

MRobot

姓名：孟祥 学号：3120190163

1. 作业类型

高阶作业。

2. 涉及到的知识

1	陈老师：多连杆动力学建模
2	李老师：电机模型及三环 PID 调试
3	余老师：运动控制

3. 研究目标和研究内容

研究目标：综合运用三位老师讲述的知识，对结合多连杆模型和电机减速器模型的三维机械臂“MRobot”进行轨迹规划和运动控制，实现期望的机器人运动。

研究内容：主要包括以下几个方面：三自由度空间机械臂建模，正运动学与逆运动学求解，Lagrange 动力学推导；8526 电机模型建模，利用 PID 控制器调节电机模型的电流环、速度环和位置环；机器人的运动轨迹规划，以及对机器人的运动进行控制。

4. 实现方法

（一）MRobot 三自由度空间机械臂建模及运动学与动力学推导

MRobot 的模型如图 1 所示，是串联的三连杆空间机械臂，理论上它可以到达其运动范围内的空间内的任意一点。因此，可以通过对机器人末端点的轨迹规划使机械臂完成期望的动作。本项目也是基于此完成期望的运动动作。MRobot 的具体推导过程见项目代码的 MRobot_Derive.m 文件。主要借鉴了陈老师在上课时推导平面二连杆机器人的运动学和动力学的方法，并在此基础上加入另外两个方向的自由度，使机器人具备三维运动的特性。下面对主要的推导过程进行说

明。

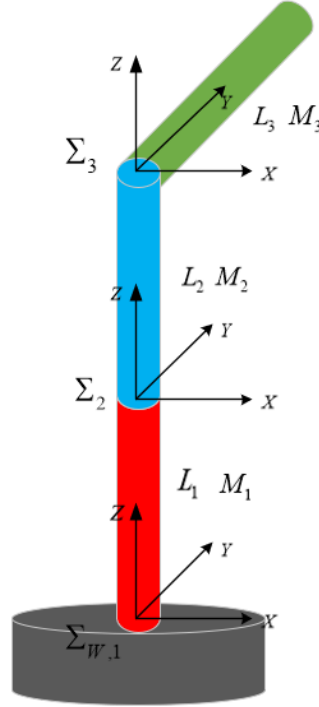


图 1 MRobot 模型及坐标系配置

给出正运动学与逆运动学的推导公式以及 Lagrange 动力学的推导过程：

推导前说明：Link1 绕 Z 轴转动 q_1 ，Link2 绕 Y 轴转动 q_2 ，Link3 绕 X 轴转动 q_3 。

(1) 正运动学 (Forward Kinematics)

主要用到的是末端点的位置，因此只给出末端点与转动角度之间的关系，机械臂的任一点在世界坐标系 (World Frame) 的位置也可以类比得到，代码参考项目代码 FK.m 文件。

$$\vec{r}_{O \rightarrow E3} = \begin{bmatrix} L_2 \cos(q_1) \sin(q_2) - L_3 (\cos(q_3) \sin(q_1) - \cos(q_1) \sin(q_2) \sin(q_3)) \\ L_3 (\cos(q_1) \cos(q_3) + \sin(q_1) \sin(q_2) \sin(q_3)) + L_2 \sin(q_1) \sin(q_2) \\ L_1 + L_2 \cos(q_2) + L_3 \cos(q_2) \sin(q_3) \end{bmatrix}$$

得到末端点的正运动学后，可以利用

$$\frac{d}{dt} \vec{r}_{O \rightarrow E3} = J \dot{\vec{q}}$$

得到末端点的 Jacobian 矩阵：

$$J = \begin{bmatrix} -L_3 (\cos(q_1) \cos(q_3) + \sin(q_1) \sin(q_2) \sin(q_3)) - L_2 \sin(q_1) \sin(q_2) & \cos(q_1) \cos(q_2) (L_2 + L_3 \sin(q_3)) & L_3 (\sin(q_1) \sin(q_3) + \cos(q_1) \cos(q_3) \sin(q_2)) \\ L_2 \cos(q_1) \sin(q_2) - L_3 (\cos(q_3) \sin(q_1) - \cos(q_1) \sin(q_2) \sin(q_3)) & \cos(q_2) \sin(q_1) (L_2 + L_3 \sin(q_3)) & L_3 \cos(q_3) \sin(q_1) \sin(q_2) - L_3 \cos(q_1) \sin(q_3) \\ 0 & -\sin(q_2) (L_2 + L_3 \sin(q_3)) & L_3 \cos(q_2) \cos(q_3) \end{bmatrix}$$

Jacobian 矩阵构建了点速度与关节速度的内在联系，并且在之后的逆运动学求解时也会用到。

（2）逆运动学（Inverse Kinematics）

对机械臂末端的轨迹进行规划后，如何将末端点的轨迹映射到关节角的轨迹，这里需要进行逆运动学的求解。逆运动学的求解方法有解析法与数值法两种。MRobot 的逆运动学采用数值求解方法。我的理解是，数值求解法本质上是一种利用末端位置误差不断迭代，不断接近期望目标位置的关节角度求解方法。具体代码参考项目代码的 NUM_IK.m 文件。

得到末端的位置误差

$$\Delta P = P_{ref} - P_{act}$$

通过判断位置误差的 2 范数的大小，当范数小于容许误差（Tolerance）时，则认为目前的关节角度满足要求，作为期望的关节位置；而当范数大于容许误差时，便进行下面的循环：

```
while norm(P_ref - P_act) > tol
    J = Get_Jacobian_M(Q_temp, model);
    delta_P = P_ref - P_act;
    Q_cal = Q_temp + pinv(J) * delta_P;
    [~, ~, P_act] = FK(Q_cal, model);
    Q_temp = Q_cal;
end
```

此过程不断迭代更新关节角度的值，直到位置误差小于容许误差，跳出该循环，将此时求出的关节角度当作期望的关节角度。

（3）Lagrange 动力学推导

Lagrange 动力学具体的推导过程参考项目代码 MRobot_Derive.m 文件。利用关节角度 $q = [q1; q2; q3]$ 和角速度 $dq = [dq1; dq2; dq3]$ 以及机械臂的机械参数得到机械臂的动能 KE 和势能 PE 后，利用 Lagrange 方程得到：

$$M\ddot{q} + C(q, \dot{q}) + G(q) = \tau$$

通过上式可以得到机器人角加速度与关节力矩之间的关系，可以利用逆动力学求解出机器人在对应关节位置的关节力矩，将该力矩作为前馈加载到电机端，进行电机控制。当然，在建模之后，需要利用机器人的动力学验证建模的准确性，这时可以不加力矩让机器人自然运动，通过观察机器人总机械能的变化情况来判

断机器人建模的准确性，具体的建模验证过程可以参考项目代码中的 My_Robot 文件夹代码。

以上便是 MRobot 整个的建模求解过程，这是整个机器人项目的基础。

（二）8526 电机建模和三环 PID 调节

电机建模部分主要参考目前实验室使用的 8526 电机模型的机械参数，对电机模型进行建模，并基于该电机模型依次对电流环、速度环和位置环进行 PID 调节，最后使位置环比较好的跟随期望的位置。

（1）8526 电机建模

主要参考李老师课上讲述的电机方程：

$$\begin{cases} Li + Ri + e = U \\ e = K_e \omega \\ \tau = K_t i \end{cases}$$

将机器人逆动力学算得的关节力矩当作负载 τ_L 加到电机端，并将电机方程改写为状态方程表达式，状态向量为 $x = [\theta; \omega; i]$ 可以得到：

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & \frac{K_t}{J_m} \\ 0 & \frac{-K_e}{L} & \frac{-R}{L} \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{L} \end{bmatrix} U + \begin{bmatrix} 0 \\ \frac{1}{J_m} \\ 0 \end{bmatrix} \tau_L$$

得到电机的状态方程表达式，通过电机电流环、速度环和位置环的三环 PID 控制，刷新电机的状态，进而带入到电机状态方程中不断更新电机的状态向量，最终使电机能够在一个位置环周期内达到期望的位置角度。

（2）电机三环 PID 调节

电机的电流环、速度环和位置环三环由电流环开始依次进行调节。具体的 PID 参数参考项目代码的 MotorController 文件夹，这里针对调节思路和一些注意事项进行总结。

PID 调节花了我两天的时间才算完全调通，并通过一个简单的单摆系统测试了电机位置跟随的效果。首先从电流环调起，这部分涉及电机模型的所有参数，主要就是要将电流环调节好，这样速度环和位置环的调节真的事半功倍。这部分

的调节过程需要注意单位的一致性，因为单位的不一致导致浪费了几个小时的时间死活调不出一个期望的曲线，心态简直要崩了。单位修改一致之后，简单对 PID 参数进行调节便得到了比较理想的曲线。因此，单位一定要注意。至于速度环和位置环的调节主要参考电流环的思路。在程序编写的过程中，由于不同的环具有不同的刷新频率，因此需要注意匹配相应的刷新次数。

最后，通过一个简单的单摆系统对电机模型进行了位置跟随测试，具体的测试代码见项目代码的 `MotorTestProg` 文件夹，结果表明电机能够很好地跟随目标轨迹，证明电机的三环 PID 调节是有效的。

（三）MRobot 运动轨迹规划和运动控制

为了让机器人能够按照期望的动作运动，首先需要对机器人的运动轨迹进行规划，规划完成后，利用逆运动学和逆动力学求解出对应的关节轨迹与关节力矩，进而对机器人的运动进行反馈控制，使机器人能够按照期望的轨迹运动。

（1）轨迹规划

MRobot 机器人的轨迹规划程序参考项目代码的 `TrajPlanning.m` 文件，目标动作是使机器人的末端能够从初始位置运动到期望的动作位置，随后开始画出一个“M”字样，完成动作后从动作结束位置运动到初始位置。

有了期望的运动后，可以将机器人末端的轨迹直接转换成代码，利用插值等方法插值出一条比较平滑的曲线。这里需要注意的是：机器人的运动范围，规划的轨迹不能超过机器人的运动范围；另外，在运动范围内，任然需要注意尽量避免关节轨迹出现奇异，否则会导致数值求解逆运动学时无解。虽然轨迹规划看起来很容易，但实操起来确实还是有很多地方值得注意。另外，考虑到实极机器人在运动过程中都会有关节限位和一些其他的环境约束和自身的机械约束，这无疑又增加了轨迹规划的难度，本项目为了简单起见，没有加入关节的限制和环境约束。

（2）运动控制

由于在电机控制时加入了机器人的逆动力学求解的力矩作为电机的负载，一定程度上可以看作是动力学前馈的作用，因此只需要对机器人关节施加简单的 PD 控制便可以使机械臂能够比较好的跟随目标轨迹，本项目便是利用前馈+反馈的控制思路对关节端进行控制。具体的反馈控制参考 `JointController.m` 文件。

5. 结果和分析

(1) MRobot 动力学建模的准确性验证:

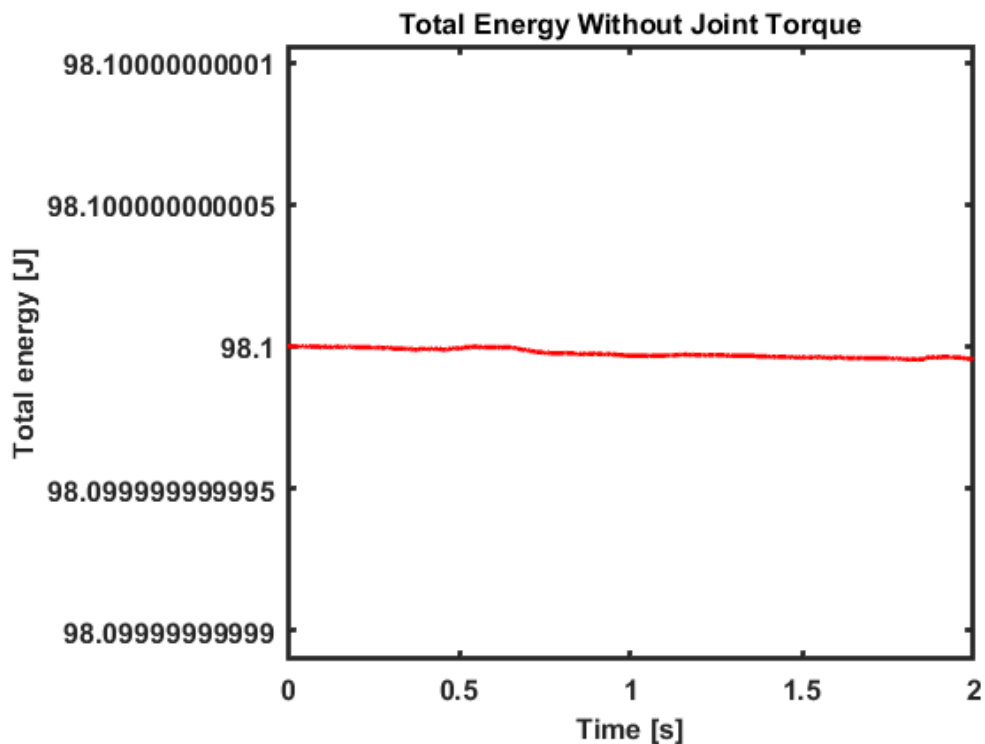


图 2 无关节力矩情况下的系统总能量曲线

在关节未施加力矩的情况下，机器人从初始状态受重力作用进行运动，机械能守恒。由图 2 可知，机器人的总能量可以看作是守恒的，因此验证机器人建模是准确的。机器人的运动学和动力学因此也是正确的。

(2) 电机 PID 调节后位置跟随有效性分析——单摆模型测试

利用单摆系统对电机模型的位置跟随效果进行测试。首先对单摆系统进行建模，因为只是测试程序，所以模型很简单，但是可以用来验证位置跟随的有效性。

图 3 为单摆测试系统的位置跟随曲线，可以看出 PID 三环调节之后的 8526 电机应用在单摆系统上能够比较好地跟随参考位置曲线，因此可以证明电机模型建模的准确性和 PID 参数调节的有效性。

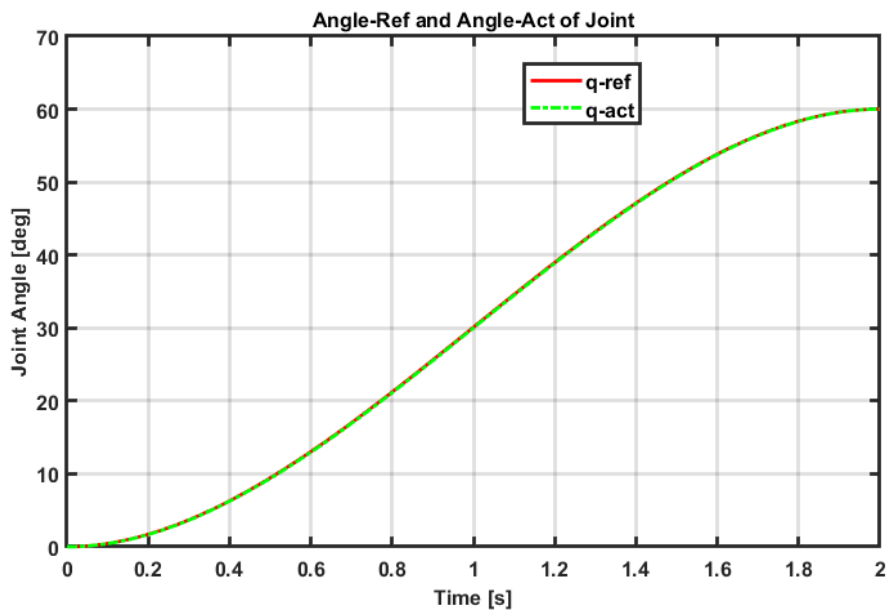


图 3 单摆模型位置跟随曲线

(3) MRobot 轨迹规划与运动控制结果分析

对运动轨迹进行规划后，利用逆运动学数值求解方法可以得到参考的关节角度。利用关节角度，角速度与角加速度可以利用逆动力学实时算出机器人的关节力矩，将关节力矩作为前馈力矩加在电机端，并结合小参数 PD 反馈控制，最终使得关节端能够比较好的跟随参考的关节轨迹。详细的机械臂轨迹跟随情况可以参考项目代码中的 MRobot_Drawing.avi 视频文件。这里给出三个关节的参考角度与实际角度曲线以及实际角度与参考角度的误差曲线。

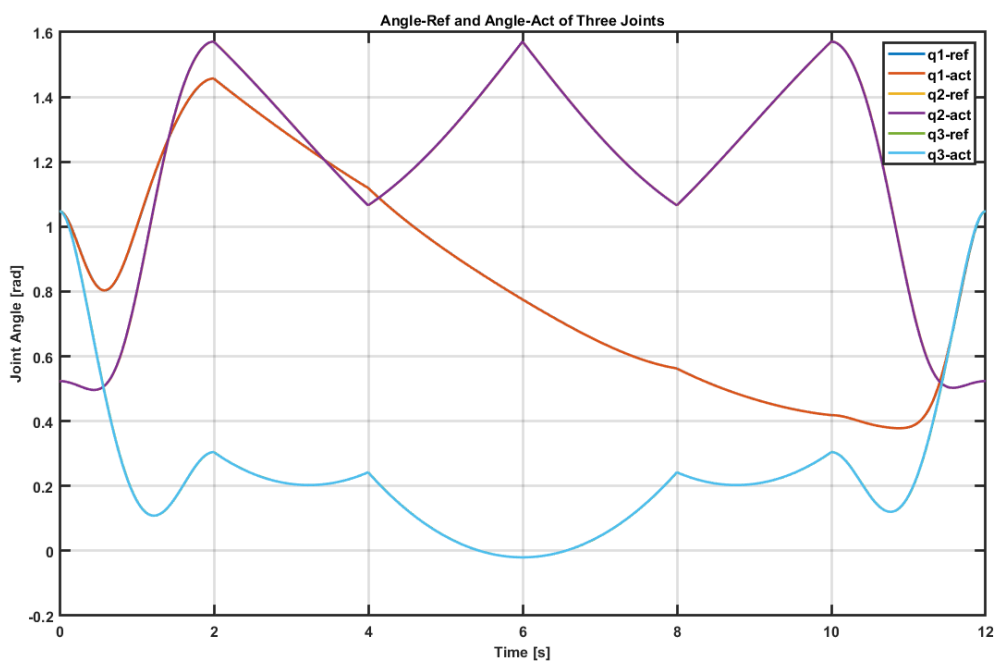


图 4 三个关节的参考角度与实际角度曲线

可以看出实际的关节角度很好地跟随了参考关节角度(图中实际角度曲线覆盖了参考关节角度曲线),基本上可以视为机器人能够按照期望的动作进行运动。进一步分析关节角度误差的情况,可以更进一步证明机器人轨迹跟随与运动控制的有效性,图5为关节角度误差变化曲线,可以看出关节角度误差始终在期望的容许误差范围内,因此可以说明机器人能够很好地按照期望的动作进行运动。这样,只需要对机器人的运动轨迹进行更改、规划,便可以得到其他的动作。

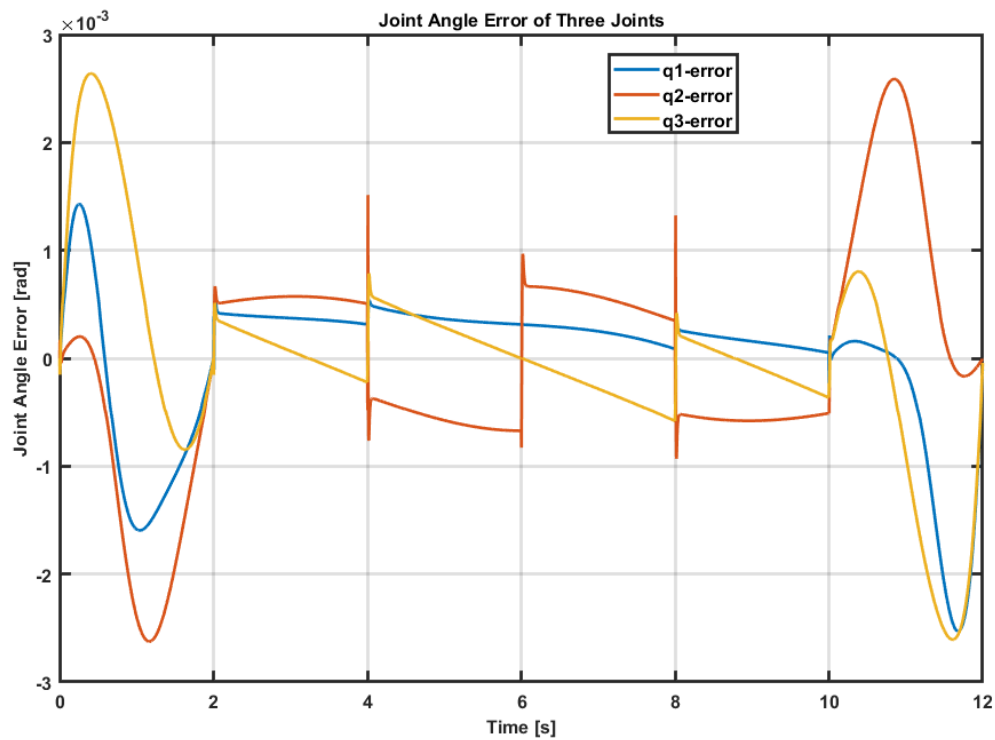


图5 三个关节的角度误差