



7 动力学仿真实验

7.1 机械臂动力学简介

在《机器人原理》这门课中，最不好讲的就是动力学。

动力学是用来干什么的呢？按理来说，如果我们的机器人有了运动学正解、逆解以及轨迹规划控制，已经足够用来控制机械臂进行作业了。因为一般情况下，我们的机械臂对于需要进行的作业，末端负载会在一个非常安全的保障系数内，不会由于末端负载过重导致运动的速度、加速度偏离作业要求太大距离。这样的作业要求也不对末端的力有严格的控制要求，所以我们一般也不太关心动力学的问题。但如果要进行打磨、切削等作业，我们是需要重点关注这个动力学问题的。

前面我们所学的运动学，我们关心了关节转角、速度、加速度甚至加加速（jerk），这个实际对应到每个关节的控制上就是，针对每个关节电机的位置（转动弧度）、转速、加速度（对应力矩，也就是电流环）的三环控制上。

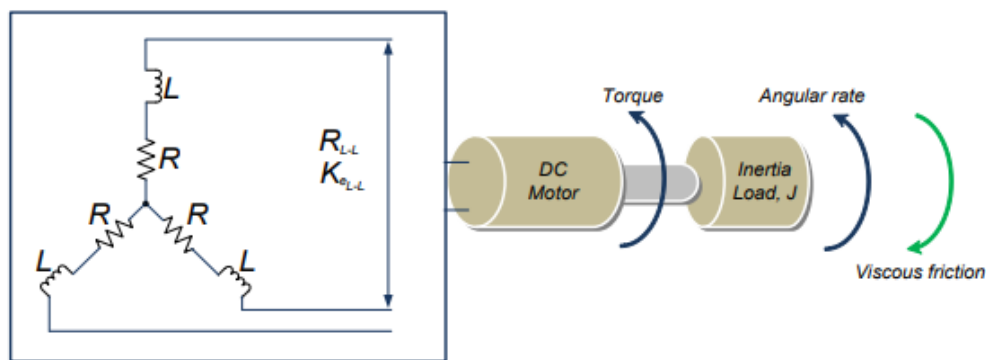


图 7.1 无刷直流电机原理图

无论是传统的机械臂也好，还是目前模块化关节臂也好，每个关节的电机控制都需要实现位置、速度、电流的三环控制，只不过前面我们做运动学实验的时候，往往只做了静力学相关的测试（同步运动到指定位置即可），速度、加速度没有太多关心。

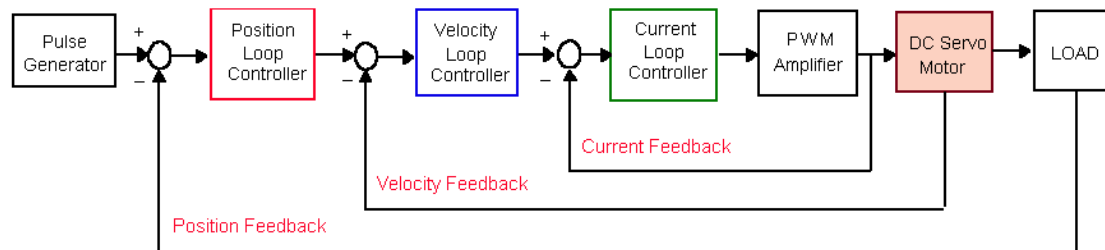


图 7.2 直流伺服电机的三环控制

关节电机的控制是机械臂控制最重要的部分，但一般机器人厂家并没有提供完全三环的反馈控制。位置反馈是必须的，速度反馈不一定提供闭环的反馈以及外部控制接口，电流环的控制就更复杂一些，一般的机器人都没有提供。



机器人动力学研究运动和受力之间的关系，所谓动力学方程，也就是机器人各关节位置、速度、加速度与关节驱动电机输出的驱动力矩之间的关系。图 7.2，我们学过，电机的输出力矩主要和电流相关。

$$T = K * \frac{U}{\omega} * I \quad 7.1$$

式 7.1 中， T 表示输出力矩， K 为系数， U 为电压， ω 为转速， I 为电流。

如果电压恒定的情况下，输出力矩和电流成正比，转速越大，力矩会降低。这个是在电机拖动与控制里面所学的最基本原理，有关机器人单关节的控制，有些教材里面都有讲，本章这里我们不细说，总之我们知道电流控制对于要实现的力矩控制，包括没有力矩传感器的情况下，要实现力的感知，是非常重要的。

前面我们介绍的运动学，有正解和逆解之分，同样，动力学也有正逆解之分。所谓**正动力学**，也就是知道了各关节的驱动力矩，它会得到什么样的响应（各关节位置、速度、加速度），这个主要是从牛顿第二定律，一个质点的运动方程出发，得到一个欧拉-拉格朗日方程来求解，这个正解求出来，一般用于**仿真**，看看力矩控制的效果如何，由于实际的作业过程中，我们最重要的一环是位置控制，所以并不能用电流控制环来完全控制关节的运动特性，电流环的控制只起到辅助控制的作用。**动力学逆解**，则是分析机器人沿着期望的轨迹运动（关节角度、速度、加速度已知）时，关节驱动电机所需要的驱动力或力矩，当然这个力矩是经过减速机之后的综合输出力矩，把这个转换成**电流反馈控制**，实现机器人精确的运动控制，特别是改善其动力学特性。

7.2 AUBO-i5 动力学仿真

动力学比较复杂，一般的教材只简单介绍一下拉格朗日法和牛顿-欧拉法，比较深一点的可能会介绍高斯(Guass)法、凯恩(Kane)法、旋量法以及罗伯逊-威登伯格(Robson Wittenburg)法等内容。

机器人动力学的基础基于刚体动力学，有关动能、势能、完整约束、虚功、达朗贝尔原理、惯量等基础请大家复习本科所学的理论力学相关基础。

动力学比较复杂，我们先介绍牛顿-欧拉法解逆动力学，再介绍拉格朗日法解正动力学的实验。

1) AUBO-i5 动力学建模

本来想采用 gazebo 来仿真动力学的，但经过测试好像很多东西还没法弄，可能我自己掌握的也不是很好，这个以后我学习好了再移植。所以本章的实验目前为止，我们停留在 matlab 加 robot toolbox（自行阅读说明，虽然比较简陋）基础上了。想模型逼真一点可以采用 vrep 加 simulink。

Peter coker 的 robot toolbox 相信大家都知道，下面我们直接开始用它给



aubo-i5 建模。

注：由于遨博目前暂无官方公布的动力学参数，所以本实验教程暂时参考 UR 官方公布的参数，等遨博公布这些参数后，我们可以把动力学参数换过来。

UR5							
Kinematics	theta [rad]	a [m]	d [m]	alpha [rad]	Dynamics	Mass [kg]	Center of Mass [m]
Joint 1	0	0	0.089159	$\pi/2$	Link 1	3.7	[0, -0.02561, 0.00193]
Joint 2	0	-0.425	0	0	Link 2	8.393	[0.2125, 0, 0.11336]
Joint 3	0	-0.39225	0	0	Link 3	2.33	[0.15, 0.0, 0.0265]
Joint 4	0	0	0.10915	$\pi/2$	Link 4	1.219	[0, -0.0018, 0.01634]
Joint 5	0	0	0.09465	$-\pi/2$	Link 5	1.219	[0, 0.0018, 0.01634]
Joint 6	0	0	0.0823	0	Link 6	0.1879	[0, 0, -0.001159]

图 7.3 UR5 的运动学与动力学参数

(<https://www.universal-robots.com/how-tos-and-faq/faq/ur-faq/parameters-for-calculations-of-kinematics-and-dynamics-45257/>)

建模代码如下：

```
%% 遨博 i5 动力学建模与仿真
```

```
close all
```

```
clear all
```

```
% using peter coker robot toolbox to build aubo-i5 model : Modified D-H
```

```
d1=0.098500;    d2=0;          d3=0;          d4=0.121500;
```

```
d5=0.102500;    d6=0.094000;
```

```
a1=0;          a2=-0.408000;    a3=-0.376000;    a4=0;          a5=0;
```

```
a6=0;
```

```
alpha1=pi/2;    alpha2= 0;      alpha3=0;      alpha4=pi/2;
```

```
alpha5=-pi/2;    alpha6=0;
```

```
% Modified DH parameter [theta d a alpha sigma(0:revolute 旋转关节  
1:prismatic 平移关节)]
```

```
% shoulder_link 建模
```

```
linkSym(1) = Link([0 d1 a1 alpha1 0]); %d a 单位是米,alpha 单位是弧度
```

```
linkSym(1).m=3.7; %连杆质量
```

```
linkSym(1).r=[0,-0.02561, 0.00193]; %连杆质心坐标
```

```
linkSym(1).G=1; % 减速比 这里我们简单设置为 1, 认为电机和减速机一体
```

```
linkSym(1).I=[0.0084 0 0; 0 0.0064 0; 0 0 0.0084]; %连杆惯性矩阵
```

```
% upperArm_link 建模
```

```
linkSym(2) = Link([0 d2 a2 alpha2 0]); %d a 单位是米,alpha 单位是弧度
```

```
linkSym(2).m=8.3930; %连杆质量
```

```
linkSym(2).r=[0.2125, 0, 0.11336]; %连杆质心坐标
```

```
linkSym(2).G=1; % 减速比 这里我们简单设置为 1, 认为电机和减速机一体
```

```
linkSym(2).I=[0.0078 0 0; 0 0.2100 0; 0 0 0.2100]; %连杆惯性矩阵
```

```
% foreArm_link 建模
```



```
linkSym(3) = Link([0 d3 a3 alpha3 0]);
linkSym(3).m=2.33;
linkSym(3).r=[0.15, 0, 0.0265];
linkSym(3).G=1;
linkSym(3).I=[0.0016 0 0; 0 0.0462 0; 0 0 0.0462];

% wrist1_link 建模
linkSym(4) = Link([0 d4 a4 alpha4 0]);
linkSym(4).m=1.219;
linkSym(4).r=[0, -0.0018, 0.01634];
linkSym(4).G=1;
linkSym(4).I=[0.0016 0 0; 0 0.0016 0; 0 0 0.0009];

% wrist2_link 建模
linkSym(5) = Link([0 d5 a5 alpha5 0]);
linkSym(5).m= 1.219;
linkSym(5).r=[0, -0.0018, 0.01634];
linkSym(5).G=1;
linkSym(5).I= [0.0016 0 0; 0 0.0016 0; 0 0 0.0009];

% wrist3_link 建模
linkSym(6) = Link([0 d6 a6 alpha6 0]);
linkSym(6).m= 0.1897;
linkSym(6).r=[0, 0, -0.001159];
linkSym(6).G=1;
linkSym(6).I= [0.0001 0 0; 0 0.0001 0; 0 0 0.0001];

%build aubo-i5 serial manipulator model
AUBOi5 = SerialLink(linkSym,'name','AUBO-i5','qlim',[0 (175/180)*pi;0
(175/180)*pi;0 (175/180)*pi;0 (175/180)*pi;0
(175/180)*pi]);
AUBOi5.payload(5,[0,0,0.1]) % 最大负载 5kg,在末端关节坐标系下,坐标为[0,0,0.1]
处,添加质量为 5kg 的负荷
base=0;
AUBOi5.base=transl([0 0 base]);
AUBOi5.plot([0 0 0 0 0 0]),axis equal
return
```

(找到了一个 notepad++的 64 位 NPP_Export 插件,贴出来代码确实好看多了。)

我们首先启动 robot toolbox:

```
>> startup_rvc
```



然后我们再运行刚才的动力学建模文件，看看 matlab 建的模型如何，见图 7.3。

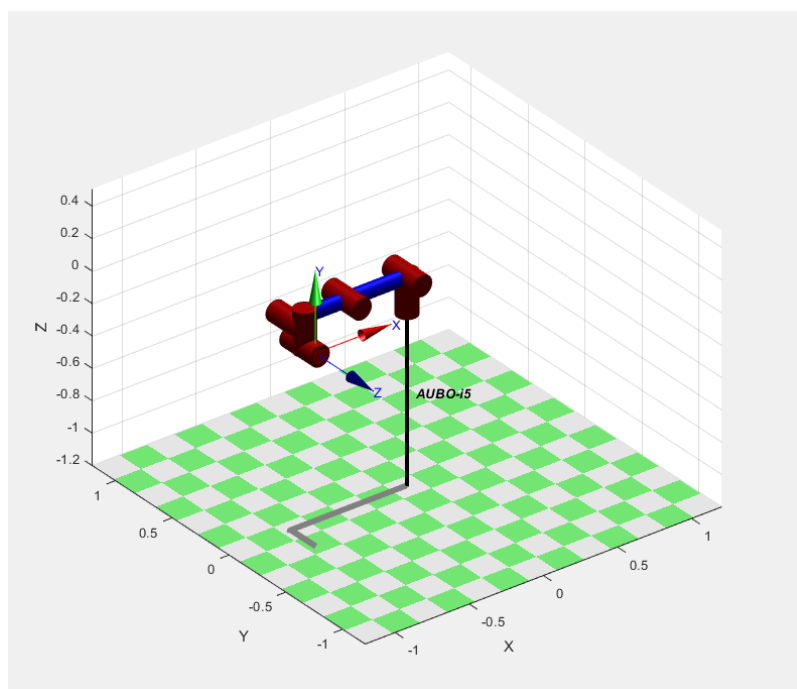


图 7.3 robot toolbox 加载 aubo-i5 动力学模型图

模型不是很好看，但简单实用。如果想要更逼真一点的效果，建议用 matlab+vrep, 这个方面实验室有同学在做，包括 simulink 留给你们自己后面实现。

接下来我们再显示一下连杆的动力学特性：

%% 显示连杆的动力学特性

```
AUBOi5.dyn(); %mass, centre of mass, inertia, gear ratio, motor inertia
and motor friction
```

```
return
```

简单一点，我们把上面代码编在一个 m 文件里，单击运行本节，可以得到 6 个连杆的动力学特性值：

```
Link 1::Revolute(std): theta=q, d=0.0985, a=0, alpha=1.5708, offset=0
m   = 3.7
r   = 0           -0.02561    0.00193
I   = | 0.0084    0           0           |
```



		0	0.0064	0	
		0	0	0.0084	

Bm = 0

Tc = 0 (+) 0 (-)

G = 1

qlim = 0.000000 to 3.054326

Link 2::Revolute(std): theta=q, d=0, a=-0.408, alpha=0, offset=0

m = 8.393

r = 0.2125 0 0.1134

I	=		0.0078	0	0	
			0	0.21	0	
			0	0	0.21	

Bm = 0

Tc = 0 (+) 0 (-)

G = 1

qlim = 0.000000 to 3.054326

Link 3::Revolute(std): theta=q, d=0, a=-0.376, alpha=0, offset=0

m = 2.33

r = 0.15 0 0.0265

I	=		0.0016	0	0	
			0	0.0462	0	
			0	0	0.0462	

Bm = 0

Tc = 0 (+) 0 (-)

G = 1

qlim = 0.000000 to 3.054326

Link 4::Revolute(std): theta=q, d=0.1215, a=0, alpha=1.5708, offset=0

m = 1.219

r = 0 -0.0018 0.01634

I	=		0.0016	0	0	
			0	0.0016	0	
			0	0	0.0009	

Bm = 0

Tc = 0 (+) 0 (-)

G = 1

qlim = 0.000000 to 3.054326

Link 5::Revolute(std): theta=q, d=0.1025, a=0, alpha=-1.5708, offset=0

m = 1.219

r = 0 -0.0018 0.01634

I	=		0.0016	0	0	
---	---	--	--------	---	---	--



```

      | 0          0.0016      0          |
      | 0          0          0.0009      |
Bm    = 0
Tc    = 0          (+) 0          (-)
G      = 1
qlim  = 0.000000 to 3.054326

Link 6::Revolute(std): theta=q, d=0.094, a=0, alpha=0, offset=0
m      = 5
r      = 0          0          0.1
I      = | 0.0001      0          0          |
      | 0          0.0001      0          |
      | 0          0          0.0001      |
Bm      = 0
Tc      = 0          (+) 0          (-)
G        = 1
qlim     = 0.000000 to 3.054326
>>

```

需要注意的是：在上面特性中我们对于黏性摩擦系数 Bm 以及库伦摩擦系数 Tc 都没有设置。

2) AUBO-i5 逆动力学：牛顿-欧拉法

牛顿-欧拉法的逆运动学，通常采用关节空间动力学方程，一般构成如式 7.2。

$$T = I(q)\ddot{q} + H(q, \dot{q}) + G(q) + T(q, q) \quad 7.2$$

其中 $I(q)$ 为机械臂的惯性矩阵，它是一个 $n \times n$ 的正定对称矩阵； $H(q, \dot{q})$ 由离心力及科氏力组成，为一个 $n \times 1$ 的矢量； $G(q)$ 为 $n \times 1$ 的重力向量； $T(q, \dot{q})$ 为考虑摩擦（主要是黏性摩擦以及库伦摩擦）的 $n \times 1$ 维摩擦阻力向量； \ddot{q} 为关节轴加速度， \dot{q} 为关节轴转速， q 为关节角； n 为连杆个数。

我们可以按照这个牛顿-欧拉公式（7.2），求解每个关节在 t 时刻下的力矩。Peter coker 编写了一个 frne.c 文件，实现了这个函数求解，其代码里面的说明如下：

```

/**
 * \file frne.c
 * \author Peter Corke
 * \brief MEX file body
 *
 *
 * FRNE      MEX file version of RNE.M
 *
 * TAU = FRNE(ROBOT, Q, QD, QDD)
 * TAU = FRNE(ROBOT, [Q QD QDD])
 *

```



```
* where Q, QD and QDD are row vectors of the manipulator state; pos,
* vel, and accel.
*
* Returns the joint torque required to achieve the specified joint
* position, velocity and acceleration state. Gravity is taken
* from the robot object.
*
* TAU = RNE(ROBOT, Q, QD, QDD, GRAV)
* TAU = RNE(ROBOT, [Q QD QDD], GRAV)
*
* GRAV overrides the gravity vector in the robot object.
*
* An external force/moment acting on the end of the manipulator may
* also be specified by a 6-element vector [Fx Fy Fz Mx My Mz].
*
* TAU = RNE(ROBOT, Q, QD, QDD, GRAV, FEXT)
* TAU = RNE(ROBOT, [Q QD QDD], GRAV, FEXT)
*
*/
```

有兴趣的话，大家可以把它翻译成 C++或其他格式代码，放到自己的工程里面用，注意版权啊：GNU Lesser General Public License。

本次实验我不打算自己编这个函数了，我准备直接在 matlab 里面来求解逆动力学。

要求逆动力学，我们首先要已知作业路径的一系列参数： \ddot{q} -关节轴加速度， \dot{q} -关节轴转速， q -关节角，然后还需要知道末端给定的负载，至于惯性矩阵、离心力矩阵、重力矩阵这个前面已经设置好。

下面的代码节展示了如何用 5 次多项式规划一段圆弧轨迹，并让 aubo-i5 沿着轨迹运动，画出一条圆弧作业轨迹。

```
% 5 次多项式轨迹规划
t = [0:.056:2];
qinit = [0,-pi/4,0,0,0,0]; %关节空间起始位置
qtargt =
[(65/180)*pi,(-55/180)*pi,(-35/180)*pi,(15/180)*pi,(10/180)*pi,(5/180)*pi]; %目标位置
% T0 = AUBOi5.fkine(qinit); %正解求位姿矩阵
% Tg = AUBOi5.fkine(qtargt); %正解求位姿矩阵
figure(1); hold on;
AUBOi5.plot(qinit); %让 aubo-i5 运动到起始点
[q,qd,qdd] = jtraj(qinit, qtargt, t); %5 次多项式轨迹规划

T = AUBOi5.fkine(q); %正运动学求解函数
% AUBOi5.plot(q); %让 aubo-i5 在图 1 上动起来
```




```
AUBOi5.animate(q);
JTA=transl(T);
plot2(JTA,'r'); % 用红色点绘制出轨迹
hold off
return
```

运行上面代码节，可以看到一段动画，机器人画出一段圆弧轨迹。如图 7.4 所示。

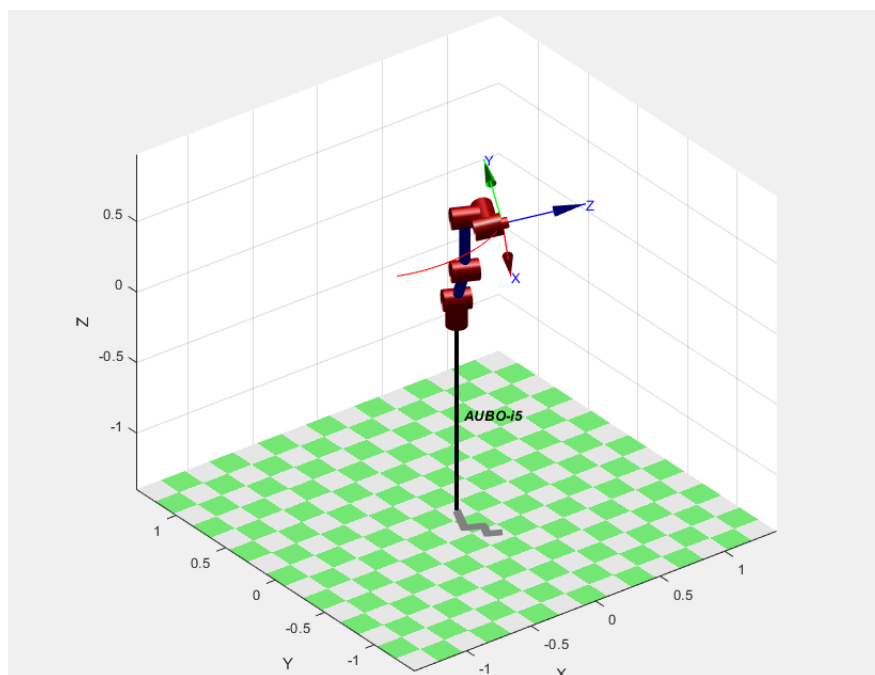


图 7.4 5 次多项式轨迹规划

现在我们要求 i5 在运行这段圆弧轨迹作业时，它的各关节的力矩情况如何。

下面这段代码节，我们演示了在两种负载情况下，这个圆弧作业轨迹的关节力矩变化情况。

```
%% 逆动力学求解
figure(2);
plot(t, q(:,1:6)); xlabel('Time (s)'); ylabel('Joint rad');
legend_h = legend('Shoulder joint','UpperArm joint','ForeArm
joint','Wrist1 joint',...
'Wrist2 joint','Wrist3 joint','Location','NorthEastOutside');
title('关节角变化图');

AUBOi5.payload(1,[0,0,0.1]) % 在末端关节坐标系下，坐标为[0,0,0.1]处，添加质
量为 1kg 的负荷
taug = AUBOi5.rne(q,qd,qdd); %逆动力学求解
figure(3);
plot(t, taug(:,1:6)); xlabel('Time (s)'); ylabel('Gravity torque (Nm)');
```



```
legend_h = legend('Shoulder link','UpperArm link','ForeArm link','Wrist1  
link',...  
    'Wrist2 link','Wrist3 link','Location','NorthEastOutside');  
title('各关节力矩变化图 (负载 1kg)');  
  
AUB0i5.payload(5,[0,0,0.1]) % 在末端关节坐标系下, 坐标为[0,0,0.1]处, 添加质  
量为 5kg 的负荷  
taug = AUB0i5.rne(q,qd,qdd); %逆动力学求解  
figure(4);  
plot(t, taug(:,1:6)); xlabel('Time (s)'); ylabel('Gravity torque (Nm)');  
  
legend_h = legend('Shoulder link','UpperArm link','ForeArm link','Wrist1  
link',...  
    'Wrist2 link','Wrist3 link','Location','NorthEastOutside');  
title('关节力矩变化图 (负载 5kg)');  
return
```

图 7.5 为这段圆弧作业的关节角规划图。

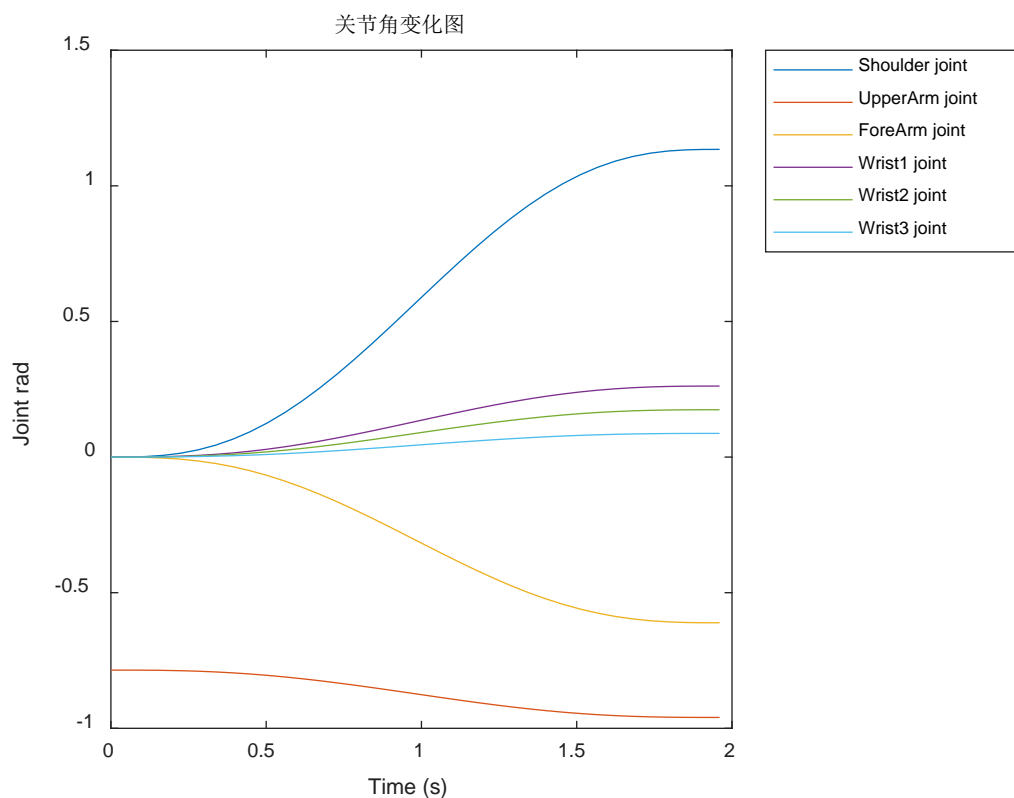


图 7.5 5 次多项式圆弧作业的关节角规划

图 7.6 为在末端关节坐标系下,坐标为[0,0,0.1]处,添加质量为 1kg 的负荷时,这段圆弧作业轨迹的各关节力矩图。

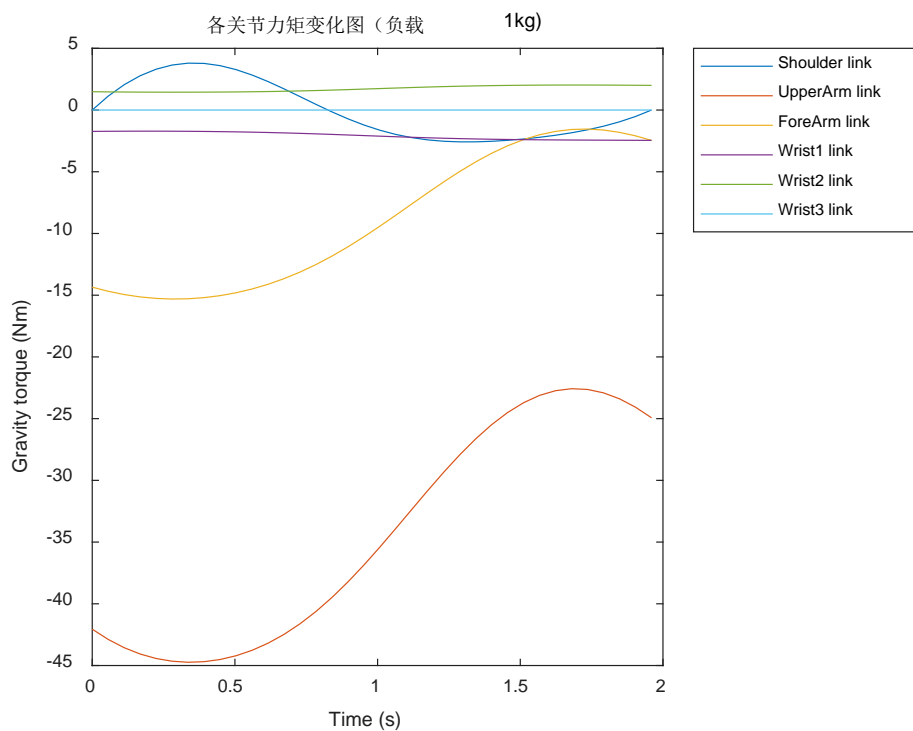


图 7.6 1kg 的负荷时, 各关节力矩图

图 7.7 为在末端关节坐标系下, 坐标为 $[0,0,0.1]$ 处, 添加质量为 5kg 的负荷时, 这段圆弧作业轨迹的各关节力矩图。

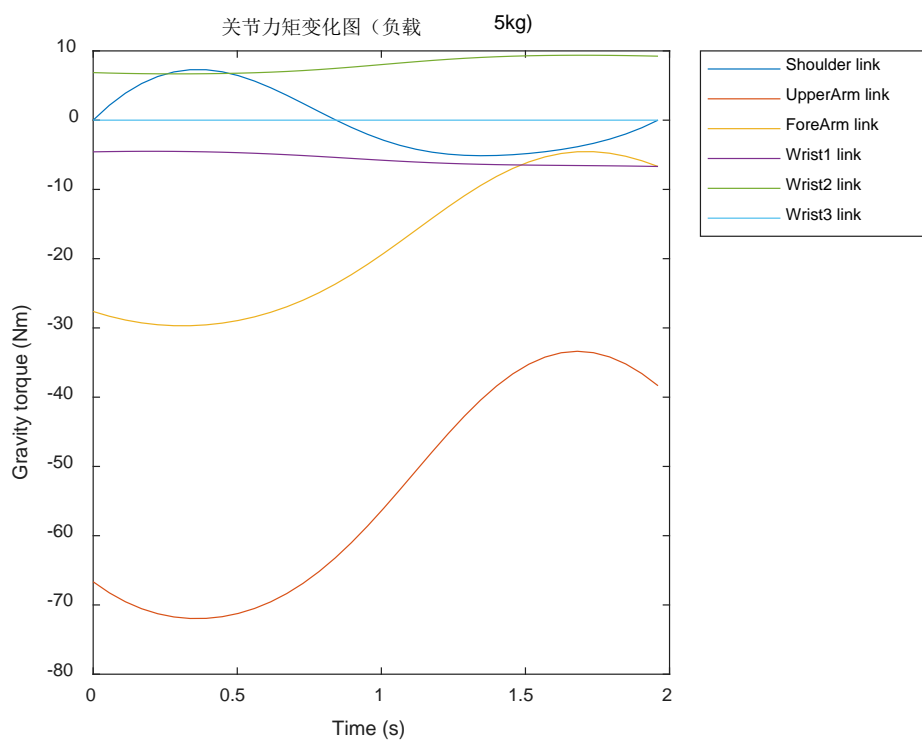


图 7.7 5kg 的负荷时, 各关节力矩图

这个力矩计算的可信度如何, 关键在于力学模型的精确程度, 如果各种惯性



质量、摩擦系数等设置正确的话，这个力矩的计算也查不到哪去，我们可以参考这个计算看我们的关节电机的电流控制是否可以满足这个要求。

3) AUBO-i5 正动力学

Robotics Toolbox 动力学正解针对串联机械臂提供了 `SerialLink.fdyn` 函数。

`[T,q,qd] = R.fdyn(T)`

该函数求解 0~T 时间范围的动力学。其说明如下：

```
%SerialLink.fdyn Integrate forward dynamics
%
% [T,Q,QD] = R.fdyn(T, TORQFUN) integrates the dynamics of the robot over
% the time interval 0 to T and returns vectors of time T, joint position
% Q
% and joint velocity QD. The initial joint position and velocity are zero.
% The torque applied to the joints is computed by the user-supplied control
% function TORQFUN:
%
%     TAU = TORQFUN(T, Q, QD)
%
% where Q and QD are the manipulator joint coordinate and velocity state
% respectively, and T is the current time.
%
% [TI,Q,QD] = R.fdyn(T, TORQFUN, Q0, QD0) as above but allows the initial
% joint position and velocity to be specified.
%
% [T,Q,QD] = R.fdyn(T1, TORQFUN, Q0, QD0, ARG1, ARG2, ...) allows optional
% arguments to be passed through to the user-supplied control function:
%
%     TAU = TORQFUN(T, Q, QD, ARG1, ARG2, ...)
%
% For example, if the robot was controlled by a PD controller we can define
% a function to compute the control
%
%     function tau = mytorqfun(t, q, qd, qstar, P, D)
%         tau = P*(qstar-q) + D*qd;
%
% and then integrate the robot dynamics with the control
%
%     [t,q] = robot.fdyn(10, @mytorqfun, qstar, P, D);
%
% Note::
% - This function performs poorly with non-linear joint friction, such
% as
% Coulomb friction. The R.nofriction() method can be used to set this
% friction to zero.
```



```
% - If TORQFUN is not specified, or is given as 0 or [], then zero torque
% is applied to the manipulator joints.
% - The builtin integration function ode45() is used.
%
% See also SerialLink.accel, SerialLink.nofriction, SerialLink.rne,
ode45.
```

这里我们先简单演示各关节附加 0 力矩，忽略库伦摩擦。

```
%% 正动力学求解-零力矩矩阵
AUBOi5 = AUBOi5.nofriction(); %忽略库伦摩擦系数
[t ,q ,qd] = AUBOi5_1.fdyn(10); %正动力学求解
[m,n] = size(q);
tau_zeros = zeros(m,n); %设置零力矩矩阵
qdd = AUBOi5.accel(q,qd,tau_zeros); %求解关节加速度
figure(5);
plot(t,qdd(:,1:6)); xlabel('Time (s)'); ylabel('Acceleration
(mm*s^-2)');
legend_h = legend('Shoulder link','UpperArm link','ForeArm link','Wrist1
link',...
'Wrist2 link','Wrist3 link','Location','NorthEastOutside');
title('零力矩求关节加速度变化图');
return
```

求解出来的关节加速度变化曲线如图 7.8 所示。

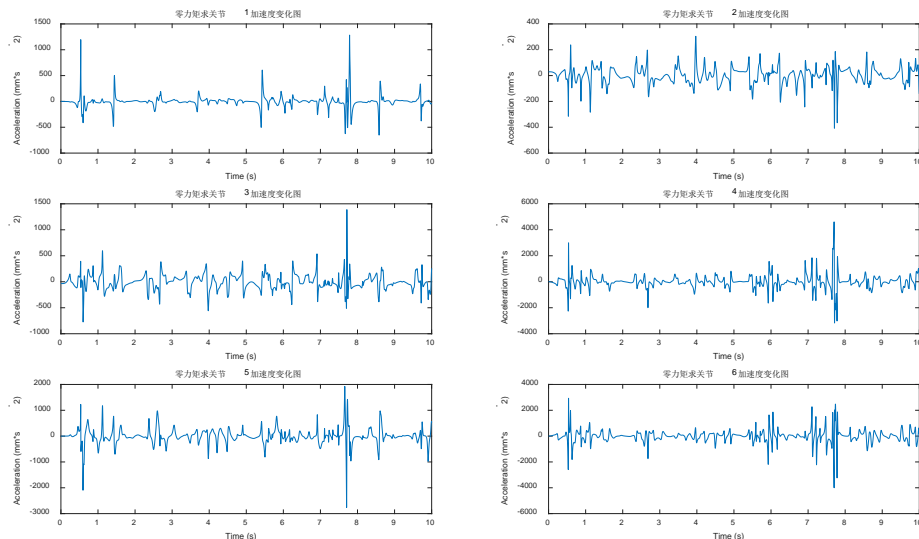


图 7.8 零力矩关节加速度变化曲线

这个零力矩产生的加速度变化曲线不太理想，可能是由于我并没有正确设置动力学参数的原因。加个随机力矩矩阵看看：

```
%% 正动力学求解-随机力矩矩阵
% 随机力矩矩阵
AUBOi5.payload(1,[0,0,0.1])
```



```
AUBOi5 = AUBOi5.nofriction(); %忽略库伦摩擦系数
[t ,q ,qd] = AUBOi5_1.fdyn(10); %正动力学求解
[m,n] = size(q);
tau_rand = 5*randn(m,n); %设置随机力矩矩阵
qdd = AUBOi5.accel(q,qd,tau_rand);%求解关节加速度
figure(7);
subplot(321),plot(t,qdd(:,1));
xlabel('Time (s)'); ylabel('Acceleration (mm*s^-2)');
title('随机力矩求关节 1 加速度变化图');
subplot(322),plot(t,qdd(:,2));
xlabel('Time (s)'); ylabel('Acceleration (mm*s^-2)');
title('随机力矩求关节 2 加速度变化图');
subplot(323),plot(t,qdd(:,3));
xlabel('Time (s)'); ylabel('Acceleration (mm*s^-2)');
title('随机力矩求关节 3 加速度变化图');
subplot(324),plot(t,qdd(:,4));
xlabel('Time (s)'); ylabel('Acceleration (mm*s^-2)');
title('随机力矩求关节 4 加速度变化图');
subplot(325),plot(t,qdd(:,5));
xlabel('Time (s)'); ylabel('Acceleration (mm*s^-2)');
title('随机力矩求关节 5 加速度变化图');
subplot(326),plot(t,qdd(:,6));
xlabel('Time (s)'); ylabel('Acceleration (mm*s^-2)');
title('随机力矩求关节 6 加速度变化图');
return
```

仿真求出各关节加速度变化如图 7.9 所示。

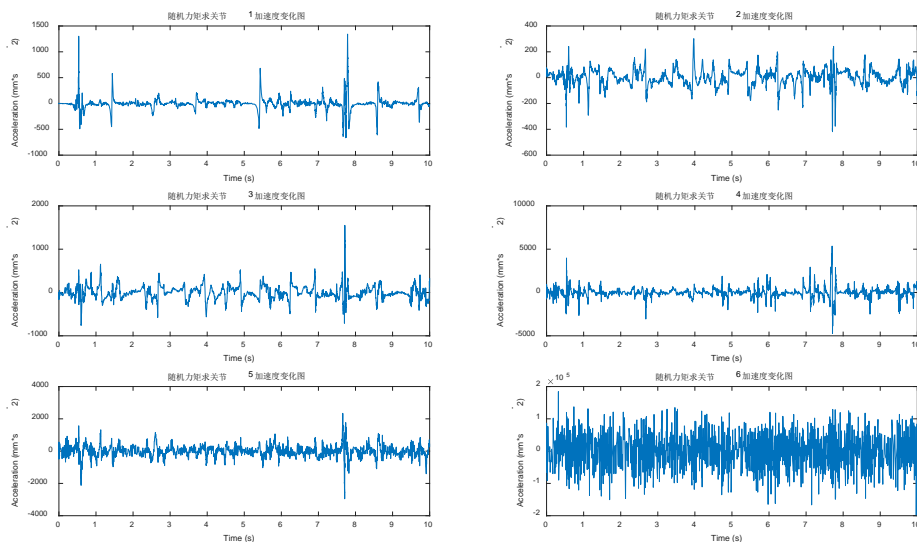


图 7.9 各关节加速度变化图

由图 7.9 可知，附加随机力矩时，各关节加速度波动非常大。



这里只是简单进行了动力学的仿真实验，数据也不太正确，如果我们要进行真正的动力学实验，可以买一个 6 维力传感器，加上加速度传感器，装在机械臂末端，去做一些实际的测试实验，指导机构的改进，以及完善机械臂的力矩控制。

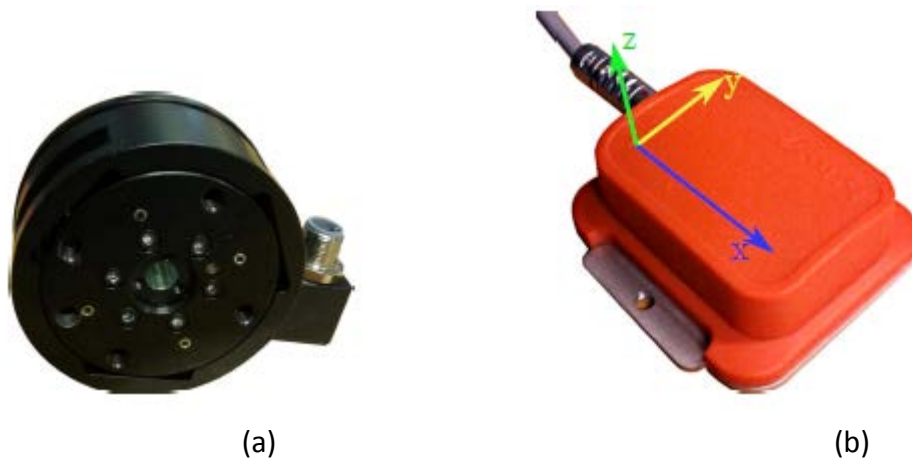


图 7.10 (a)6 维力传感器 (b)加速度传感器

7.3 本章版权及声明

本教程为武汉科技大学机器人与智能系统研究院闵华松教授实验室内部教学内容，未经授权，任何商业行为个人或组织不得抄袭、转载、摘编、修改本章内容；任何非盈利性个人或组织可以自由传播（禁止修改、断章取义等）本网站内容，但是必须注明来源。

本章内容由闵华松(mhuasong@wust.edu.cn)编写。