# 雅可比矩阵

**auto& fwd = dynamic\_cast<ForwardKinematicSolver&>(plan.model()->solverPool()[1]);**

**if (fwd.kinPos()) return -1;**

**fwd.cptJacobiWrtEE();//末端法兰盘坐标系在基座标系下的雅可比矩阵**

**auto J1 = fwd.Jf();**

更换法兰盘坐标系与基座标系，求取雅可比矩阵

# 动力学与控制

在用动力学做一些控制时，不光需要逆动力学计算的结果，还需要惯量矩阵等信息。一般动力学方程的形式为

**tau=M(q)\*ddq+C(q,dq)\*dq+G(q) (1)**

创建模型后，给模型的各个关节赋上位置、速度与加速度

**for (int i = 0; i < num; i++) {**

**model()->motionPool()[i].setMp(1);**

**model()->motionPool()[i].setMv(1);**

**model()->motionPool()[i].setMa(1);**

**}**

调用正动力学求解器，依次进行位置正解、速度正解等

**auto& fwd = dynamic\_cast<model::ForwardKinematicSolver&>(model()->solverPool()[1]);**

**if (fwd.kinPos())return -1;**

**fwd.kinVel();**

**fwd.cptGeneralInverseDynamicMatrix();**

调用逆动力学求解器

model()->solverPool().at(2).dynAccAndFce();

此后可分别获取表达式(1)中的部分项：

**auto h = fwd.h()**; //**h相当于动力学方程的C(q,dq)\*dq+G(q)**

**s\_mc(num, num, fwd.M(), fwd.nM(), M, num)**;//这里的M(num\*num)即为M\*ddq里的M

也可以将关节动力学方程转换为空间动力学方程的形式

**//以六轴机器人为例**

**double U[6\*6], tau[6], J\_inv[6\*6], tau2[6];**

**Size p[6], rank;**

**s\_householder\_utp(6, 6, fwd.Jf(), U, tau, p, rank, 1e-4);**

**s\_householder\_utp2pinv(6, 6, rank, U, tau, p, J\_inv, tau2, 1e-7);//获得雅可比逆矩阵**

**double M[6\*6], tem[6\*6];**

**s\_mc(6, 6, fwd.M(), fwd.nM(), M, 6);//这里的M即为M\*ddq里的M**

**for (int i = 0; i < 6; ++i)M[at(i, i, 6)] += m->motionPool()[i].frcCoe()[2];//加上每个轴电机和减速器的转动惯量**

**s\_mm(6, 6, 6, M, J\_inv, tem);//先右乘J\_inv**

**s\_mm(6, 6, 6, J\_inv, T(6), tem, 6, Mx, 6);//再左乘J\_inv的转置，注意矩阵转置的逆等于矩阵逆的转置**