

Robotics 2

Midterm Test – April 24, 2024

Exercise #1

Consider the 3R planar robot in the configuration \mathbf{q} shown in Fig. 1, controlled by the joint velocity $\dot{\mathbf{q}} \in \mathbb{R}^3$.

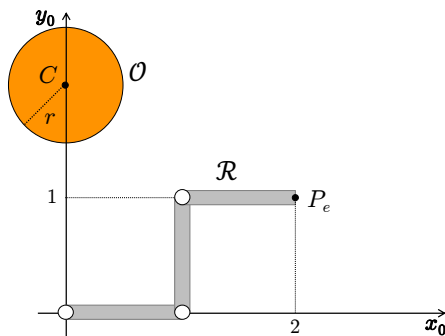


Figure 1: The 3R planar robot \mathcal{R} with a single circular obstacle \mathcal{O} in its workspace.

- i) A desired Cartesian velocity $\mathbf{v}_e \in \mathbb{R}^2$ should be imposed to the end-effector point P_e of the robot \mathcal{R} , while maximizing the distance function $H_{\text{dist}}(\mathbf{q})$ between the robot body (i.e., its kinematic skeleton) and a circular obstacle \mathcal{O} centered at $C = (0, 2)$ and of radius $r = 0.5$ m, as expressed by

$$H_{\text{dist}}(\mathbf{q}) = \min_{\mathbf{p} \in \mathcal{R}, \mathbf{o} \in \mathcal{O}} \|\mathbf{p}(\mathbf{q}) - \mathbf{o}\| = \|P_m(\mathbf{q}) - O_m\|,$$

where $P_m(\mathbf{q})$ and O_m are the closest points, respectively on the robot and on the obstacle, when the robot is in the given configuration \mathbf{q} . Determine the symbolic expression and the numerical value of the velocity command $\dot{\mathbf{q}}_{PG}$ according to the Projected Gradient (PG) method when $\mathbf{v}_e = (0 \ 1)^T$ m/s. Verify your result and compute also the resulting velocity $\mathbf{v}_m \in \mathbb{R}^2$ of the point P_m .

- ii) Suppose now that the robot \mathcal{R} has two tasks assigned, ordered by priority: the first task is to impose the same previous velocity \mathbf{v}_e to the point P_e ; the second task is to impose the velocity \mathbf{v}_m obtained in the previous item to the point P_m . Determine the symbolic expression and the numerical value of the velocity command $\dot{\mathbf{q}}_{TP}$ according to the Task Priority (TP) method. Do the joint velocities $\dot{\mathbf{q}}_{TP}$ and $\dot{\mathbf{q}}_{PG}$ have different directions in the joint space or not? Explain why. What if we choose $\mathbf{v}_m = \alpha (P_m - O_m)$, for some $\alpha > 0$, as desired velocity for P_m ?

Exercise #2

Figure 2 shows a 2R robot moving under gravity in the 3D space, with the associated Denavit-Hartenberg (D-H) frames. The following assumptions are made on the position of the center of mass and on the barycentric inertia of the two links, when these quantities are expressed in the frame attached to the respective link¹:

$${}^1\mathbf{r}_{c1} = \begin{pmatrix} r_{c1,x} \\ r_{c1,y} \\ r_{c1,z} \end{pmatrix}, \quad {}^2\mathbf{r}_{c2} = \begin{pmatrix} r_{c2,x} \\ 0 \\ 0 \end{pmatrix}, \quad {}^1\mathbf{I}_{c1} = \begin{pmatrix} I_{c1,xx} & I_{c1,xy} & I_{c1,xz} \\ I_{c1,xy} & I_{c1,yy} & I_{c1,yz} \\ I_{c1,xz} & I_{c1,yz} & I_{c1,zz} \end{pmatrix}, \quad {}^2\mathbf{I}_{c2} = \text{diag}\{I_{c2,xx}, I_{c2,yy}, I_{c2,zz}\}.$$

Note in particular that the center of mass of the first link is not on the axis of joint 1.

- i) Derive the dynamic model in the form

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau},$$

and a factorization matrix \mathbf{S} such that $\mathbf{S}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} = \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}})$ and matrix $\dot{\mathbf{M}} - 2\mathbf{S}$ is skew-symmetric.

¹All given symbolic quantities are assumed to be generically non-zero.

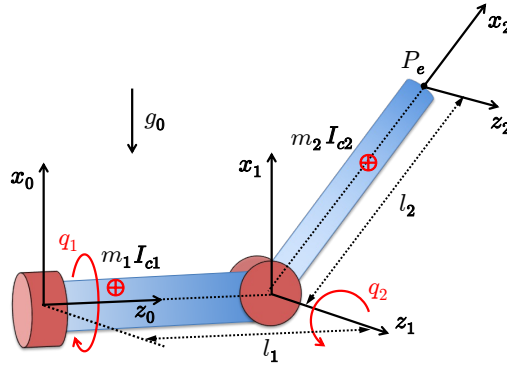


Figure 2: A 2R spatial robot with its D-H frames.

ii) Provide a minimal linear parametrization of the model

$$\mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \mathbf{a} = \boldsymbol{\tau},$$

giving the symbolic expression of the dynamic coefficients $\mathbf{a} \in \mathbb{R}^p$ and of the $2 \times p$ regressor matrix \mathbf{Y} . Assume that the acceleration of gravity g_0 is known.

iii) What is the expression of the torque $\boldsymbol{\tau}_d(t) = (\tau_{d1}(t), \tau_{d2}(t))$ needed to execute the desired motion $\mathbf{q}_d(t) = (q_{d1}(t), q_{d2}(t)) = (2t, \pi/4)$, with $t \in [0, \pi]$?

iv) When $\boldsymbol{\tau} = \mathbf{0}$, find as many as possible (unforced) equilibrium states $\mathbf{x}_e = (\mathbf{q}_e, \mathbf{0})$ of the robot —specify also the assumptions under which such equilibrium configurations \mathbf{q}_e exist.

v) Choose mechanical parameters for the links such that the gravity term $\mathbf{g}(\mathbf{q})$ vanishes for all \mathbf{q} .

vi) Under the conditions found in item v), with the robot at rest and being $\boldsymbol{\tau} = \mathbf{0}$, a force $\mathbf{F}_e \in \mathbb{R}^2$ is applied to the tip of the second link along a direction lying in the horizontal plane parallel to the plane $(\mathbf{y}_0, \mathbf{z}_0)$. Will the resulting tip acceleration $\ddot{\mathbf{p}}_e$ be in the same direction of the applied force \mathbf{F}_e or not? Elaborate your answer.

Exercise #3

With reference to the standard recursive Newton-Euler algorithm for inverse dynamics, consider the calling instruction

$$\mathbf{u} = NE_0(\mathbf{q}, \mathbf{0}, \mathbf{0})$$

for a 6R robot in a nonsingular configuration \mathbf{q} . What will be the output \mathbf{u} when the robot end-effector is i) in free space, or ii) subject to a known active wrench (force and moment) $\mathbf{F}_e \in \mathbb{R}^6$?

Exercise #4

Consider a rigid robot manipulator with n revolute joints, without external contact forces/moments and dissipative effects. The components of the motor torque $\boldsymbol{\tau} \in \mathbb{R}^n$ are bounded as

$$|\tau_i| \leq T_i, \quad \text{with } T_i \geq 5 \cdot \max_{\mathbf{q}} |g_i(\mathbf{q})|, \quad i = 1, \dots, n,$$

namely with bounds that are large enough to guarantee that the robot always sustains at least its own weight under gravity (here, with a conservative margin factor of 5). Let the robot state at time $t = t_0$ be $\mathbf{x}(t_0) = (\mathbf{q}(t_0), \dot{\mathbf{q}}(t_0))$, with a total robot energy $E(t_0) = E_0$.

i) If the robot is in a generic state $\mathbf{x}(t_0) = (\mathbf{q}_0, \dot{\mathbf{q}}_0)$ with $\dot{\mathbf{q}}_0 \neq \mathbf{0}$, choose the torque $\boldsymbol{\tau}_0 = \boldsymbol{\tau}(t_0)$ to be delivered by the motors so that the total energy E will instantaneously decrease as much as possible.

ii) Suppose now that the robot is in a state $\mathbf{x}(t_0) = (\mathbf{q}_0, \mathbf{0})$, namely **at rest** and in a configuration \mathbf{q}_0 such that $\mathbf{g}(\mathbf{q}_0) \neq \mathbf{0}$. Choose again the motor torque $\boldsymbol{\tau}_0$ so that the total energy E will decrease. *Hint:* If this is not possible, choose $\boldsymbol{\tau}_0$ so that \dot{E} will instantaneously decrease.

[210 minutes (3.5 hours); open books]

Solution

April 24, 2024

Exercise #1

The needed formulas to be evaluated at $\mathbf{q}_0 = (0, \pi/2, -\pi/2)$ are the following.

i) For the Projected Gradient (PG) method:

$$\begin{aligned}\dot{\mathbf{q}}_{PG} &= \mathbf{J}_e^\#(\mathbf{q}_0)\mathbf{v}_e + \mathbf{P}_e(\mathbf{q}_0)\nabla_{\mathbf{q}}H_{\text{dist}}(\mathbf{q}_0) = \mathbf{J}_e^\#(\mathbf{q}_0)\mathbf{v}_e + (\mathbf{I} - \mathbf{J}_e^\#(\mathbf{q}_0)\mathbf{J}_e(\mathbf{q}_0))\nabla_{\mathbf{q}}H_{\text{dist}}(\mathbf{q}_0) \\ &= \nabla_{\mathbf{q}}H_{\text{dist}}(\mathbf{q}_0) + \mathbf{J}_e^\#(\mathbf{q}_0)(\mathbf{v}_e - \mathbf{J}_e(\mathbf{q}_0)\nabla_{\mathbf{q}}H_{\text{dist}}(\mathbf{q}_0)),\end{aligned}$$

with

$$H_{\text{dist}}(\mathbf{q}_0) = \|P_m(\mathbf{q}_0) - C\| - r, \quad \nabla_{\mathbf{q}}H_{\text{dist}}(\mathbf{q}_0) = \frac{1}{2}\mathbf{J}_m(\mathbf{q}_0)\frac{P_m(\mathbf{q}_0) - C}{\|P_m(\mathbf{q}_0) - C\|},$$

where $P_m(\mathbf{q}_0) = (1, 1)$. Moreover,

$$\mathbf{v}_m = \mathbf{J}_m(\mathbf{q}_0)\dot{\mathbf{q}}_{PG}.$$

ii) For the Task Priority (TP) method:

$$\dot{\mathbf{q}}_{TP} = \mathbf{J}_e^\#(\mathbf{q}_0)\mathbf{v}_e + (\mathbf{J}_m(\mathbf{q}_0)\mathbf{P}_e(\mathbf{q}_0))^\#(\mathbf{v}_2 - \mathbf{J}_m(\mathbf{q}_0)\mathbf{J}_e^\#(\mathbf{q}_0)\mathbf{v}_e),$$

where in the first case (A) $\mathbf{v}_2 = \mathbf{v}_m$ and in the second case (B) $\mathbf{v}_2 = \left(1 - \frac{r}{\|P_m - C\|}\right)(P_m - C)$ (namely, a velocity along the direction of the gradient of the Cartesian clearance between robot and obstacle, with norm equal to $H_{\text{dist}}(\mathbf{q}_0)$).

The MATLAB code `Ex1.Midterm.m`, whose source text is given below, has been used for both items *i)* and *ii)*. The output is attached at the end of this pdf file.

The result is that $\dot{\mathbf{q}}_{TP,A} = \dot{\mathbf{q}}_{PG}$: the TP method returns in this case exactly the requested velocity \mathbf{v}_m for the robot point P_m . This is because \mathbf{v}_m is certainly realizable together with the velocity \mathbf{v}_E of the end-effector, having been defined as the natural outcome of the PG method for point P_m . On the other hand, $\dot{\mathbf{q}}_{TP,B} \neq \dot{\mathbf{q}}_{PG}$, both in direction and intensity in the joint space. In this case, the requested velocity for the second task, one that would move the closest robot point P_m in the farthest away direction from the obstacle, is not compatible with the velocity \mathbf{v}_E of the end-effector.² As a result, $\mathbf{J}_m(\mathbf{q}_0)\dot{\mathbf{q}}_{TP,B}$ will not be equal to the desired \mathbf{v}_2 in case B.

```
% Robotics 2 Midterm
% 24 April 2024
% Ex #1
%
% Projected Gradient (PG) method (with obstacle avoidance)
% and Task Priority (TP) method (with two variants)
% for a 3R planar robot

clear all, clc

disp('*** Projected Gradient and Task Priority (TP) methods for a 3R planar robot ***')
disp(' ')

syms q1 q2 q3 real
q=[q1 q2 q3]';

disp('kinematics of the 3R planar robot (links of unitary length)')
```

²This should not be unexpected for generic directions \mathbf{v}_m and \mathbf{v}_e , as the sum of the dimensions of these two tasks ($m = m_1 + m_2 = 4$) is larger than the dimension of the joint space ($n = 3$).

```

pe_simb=[cos(q1)+cos(q1+q2)+cos(q1+q2+q3);sin(q1)+sin(q1+q2)+sin(q1+q2+q3)]
pm_simb=[cos(q1)+cos(q1+q2);sin(q1)+sin(q1+q2)]

Je_simb=jacobian(pe_simb,q)
Jm_simb=jacobian(pm_simb,q)

disp('at the given configuration (as in the text figure)')

q0=[0;pi/2;-pi/2]

disp('numerical kinematics')

pe=subs(pe_simb,q,q0)
pm=subs(pm_simb,q,q0)

Je=subs(Je_simb,q,q0)
Jm=subs(Jm_simb,q,q0)

disp('clearance')

C=[0;2];r=0.5;
H=norm(pm-C)-r

disp('desired end-effector velocity')

ve=[0 1]'

disp('projected gradient (PG) method')

nablaH=0.5*Jm*(pm-C)/norm(pm-C)
nablaH=eval(nablaH)
alfa1=1 % works also for a different alfa1>0

Je_pinv=pinv(Je)
dq_PG=alfa1*nablaH+Je_pinv*(ve-alfa1*Je*nablaH)
dq_PG=eval(dq_PG)

disp('check PG solution')

ve_check=eval(Je*dq_PG)
vm=eval(Jm*dq_PG)

% TP with two cases: A & B

disp('Task Priority (TP) - case A')

Pe=eye(3)-Je_pinv*Je
vm_A=vm

dq_TP_A=Je_pinv*ve+pinv(Jm*Pe)*(vm_A-Jm*Je_pinv*ve)
dq_TP_A=eval(dq_TP_A)

disp('check TP-A solution')

ve_check=eval(Je*dq_TP_A)
vm_check=eval(Jm*dq_TP_A)

disp('Task Priority (TP) - case B')

alfa2=1 % minimum error = vm-Jm*dq_TP_B is obtained with alfa2=0.9
vm_B=alfa2*(1-(r/norm(pm-C)))*(pm-C)
vm_B=eval(vm_B)

dq_TP_B=Je_pinv*ve+pinv(Jm*Pe)*(vm_B-Jm*Je_pinv*ve)

```

```

dq_TP_B=eval(dq_TP_B)
disp('check TP_B solution')
ve_check=eval(Je*dq_TP_B)
vm_check=eval(Jm*dq_TP_B)
vm_check_error=eval(vm_B-Jm*dq_TP_B)
% end

```

Exercise #2

It is useful (though not strictly necessary) to use the moving frame algorithm for deriving the kinetic energy of the links, as well as the Denavit-Hartenberg (D-H) homogeneous transformation matrices for the potential energy due to gravity. For this, one should start from the table of D-H parameters given in Tab. 1.

i	α_i	a_i	d_i	θ_i
1	$-\pi/2$	0	l_1	q_1
2	0	l_2	0	q_2

Table 1: D-H parameters corresponding to the frames in Fig. 2.

i) The dynamic model computations provide the following terms in symbolic form:

$$\begin{aligned}
\mathbf{M}(\mathbf{q}) &= \begin{pmatrix} a_1 + a_2 \cos^2 q_2 & 0 \\ 0 & a_3 \end{pmatrix} = \\
&\begin{pmatrix} I_{c1,yy} + I_{c2,xx} + m_1(r_{c1,x}^2 + r_{c1,z}^2) + (I_{c2,yy} - I_{c2,xx} + m_2(l_2 + r_{c2,x})^2) \cos^2 q_2 & 0 \\ 0 & I_{c2,zz} + m_2(l_2 + r_{c2,x})^2 \end{pmatrix} \\
\mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) &= \begin{pmatrix} -2a_2 \dot{q}_1 \dot{q}_2 \cos q_2 \sin q_2 \\ a_2 \dot{q}_1^2 \cos q_2 \sin q_2 \end{pmatrix} = \begin{pmatrix} -a_2 \dot{q}_2 \cos q_2 \sin q_2 & -a_2 \dot{q}_1 \cos q_2 \sin q_2 \\ a_2 \dot{q}_1 \cos q_2 \sin q_2 & 0 \end{pmatrix} \dot{\mathbf{q}} = \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} \\
\mathbf{g}(\mathbf{q}) &= \begin{pmatrix} a_4 g_0 \cos q_1 \sin q_2 + a_5 g_0 \sin q_1 + a_6 g_0 \cos q_1 \\ a_4 g_0 \cos q_1 \sin q_2 \end{pmatrix} \\
&= \begin{pmatrix} -m_2(l_2 + r_{c2,x}) g_0 \cos q_1 \sin q_2 - m_1 r_{c1,x} g_0 \sin q_1 - m_1 r_{c1,z} g_0 \cos q_1 \\ -m_2(l_2 + r_{c2,x}) g_0 \cos q_1 \sin q_2 \end{pmatrix}.
\end{aligned}$$

ii) A linear parametrization of the dynamic model, $\mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \mathbf{a} = \boldsymbol{\tau}$, of dimension $p = 6$ uses the following dynamic coefficients

$$\mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{pmatrix} \begin{pmatrix} I_{c1,yy} + I_{c2,xx} + m_1(r_{c1,x}^2 + r_{c1,z}^2) \\ I_{c2,yy} - I_{c2,xx} + m_2(l_2 + r_{c2,x})^2 \\ I_{c2,zz} + m_2(l_2 + r_{c2,x})^2 \\ -m_2(l_2 + r_{c2,x}) \\ -m_1 r_{c1,x} \\ -m_1 r_{c1,z} \end{pmatrix}$$

and the 2×6 regressor matrix

$$\mathbf{Y} = \begin{pmatrix} \ddot{q}_1 & \cos^2 q_2 \ddot{q}_1 - 2 \sin q_2 \cos q_2 \dot{q}_1 \dot{q}_2 & 0 & g_0 \sin q_1 \cos q_2 & g_0 \sin q_1 & g_0 \cos q_1 \\ 0 & \sin q_2 \cos q_2 \dot{q}_1 \dot{q}_2 & \ddot{q}_2 & g_0 \cos q_1 \sin q_2 & 0 & 0 \end{pmatrix}.$$

iii) The inverse dynamics torque associated to the desired motion $\mathbf{q}_d(t) = (2t, \pi/4)$ is given by

$$\boldsymbol{\tau}_d(t) = \begin{pmatrix} a_5 g_0 + a_4 \frac{\sqrt{2}}{2} g_0 \sin 2t + a_6 g_0 \cos 2t \\ 2a_2 + a_4 \frac{\sqrt{2}}{2} g_0 \cos 2t \end{pmatrix}$$

iv) The (unforced) equilibrium configurations \mathbf{q}_e are found by solving the condition $\mathbf{g}(\mathbf{q}_e) = \mathbf{0}$. In particular, zeroing the second component of \mathbf{g} leads to the two following cases:

$$\begin{aligned} \cos q_{1,e} = 0 &\Rightarrow q_{1,e} = \{\pm \frac{\pi}{2}\} \Rightarrow q_{2,e} = \pi \pm \arccos \frac{a_5}{a_4} \quad \Leftarrow \text{ valid only if } |a_5| \leq |a_4|; \\ \sin q_{2,e} = 0 &\Rightarrow q_{2,e} = \begin{cases} 0 &\Rightarrow q_{1,e} = 2 \arctan \frac{a_4 + a_5 \pm \sqrt{(a_4 + a_5)^2 + a_6^2}}{a_6} \\ \pi &\Rightarrow q_{1,e} = \pm 2 \arctan \frac{a_4 - a_5 \pm \sqrt{(a_4 + a_5)^2 + a_6^2}}{a_6} \end{cases} \quad \Leftarrow \text{ always valid.} \end{aligned}$$

v) The mechanical conditions for having $\mathbf{g}(\mathbf{q}) \equiv \mathbf{0}$ for all \mathbf{q} are:

$$r_{c2,x} = -l_2 \Rightarrow a_4 = 0; \quad r_{c1,x} = 0 \Rightarrow a_5 = 0; \quad r_{c1,z} = 0 \Rightarrow a_6 = 0.$$

vi) With the robot at rest ($\dot{\mathbf{q}} = \mathbf{0}$), with $\mathbf{g}(\mathbf{q}) \equiv \mathbf{0}$ from the previous item, and with $\boldsymbol{\tau} = \mathbf{0}$, the dynamics in reaction to a force $\mathbf{F}_e \in \mathbb{R}^2$ applied to the end-effector only in the plane $(\mathbf{y}_0, \mathbf{z}_0)$ is

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} = \mathbf{J}_{e,yz}^T(\mathbf{q})\mathbf{F}_e, \quad \text{with } \mathbf{J}_{e,yz}(\mathbf{q}) = \frac{\partial \begin{pmatrix} p_{e,y} \\ p_{e,z} \end{pmatrix}}{\partial \mathbf{q}} = \begin{pmatrix} l_2 \cos q_1 \cos q_2 & -l_2 \sin q_1 \sin q_2 \\ 0 & -l_2 \cos q_2 \end{pmatrix}.$$

Therefore, the end-effector acceleration is computed as

$$\begin{aligned} \ddot{\mathbf{p}}_e &= (\mathbf{J}_{e,yz}(\mathbf{q})\mathbf{M}^{-1}(\mathbf{q})\mathbf{J}_{e,yz}^T(\mathbf{q}))\mathbf{F}_e = \mathbf{M}_p^{-1}(\mathbf{q})\mathbf{F}_e \\ &= \begin{pmatrix} l_2^2 (F_{e,y} ((a_1 + a_2 \cos^2 q_2) \cos^2 q_1 \cos^2 q_2 + a_3 \sin^2 q_1 \sin^2 q_2) + F_{e,z} a_3 \sin q_1 \sin q_2 \cos q_2) \\ a_3 l_2^2 \cos q_2 (F_{e,y} \sin q_1 \sin q_2 + F_{e,z} \cos q_2) \end{pmatrix}. \end{aligned}$$

where the matrix premultiplying \mathbf{F}_e is the inverse of the robot inertia matrix at the end-effector level, restricted to the plane of interest in the Cartesian space. Being this matrix non-diagonal, the direction of the acceleration $\ddot{\mathbf{p}}_e$ will differ from the direction of the applied force \mathbf{F}_e (although the scalar product $\mathbf{F}_e^T \ddot{\mathbf{p}}_e > 0$, out of singularities of the Jacobian $\mathbf{J}_{e,yz}$).

All the above computations have been done first by hand. The MATLAB code `Ex1_Midterm.m`, whose source text is given below, has been used to verify the correctness of all items from *i*) to *vi*). In the source, some extra steps useful for symbolic manipulation and further comments are preceded by `'% ADL:'`). The output is attached at the end of this pdf file.

```
% Robotics 2 Midterm
% 24 April 2024
% Ex #2
%
% Dynamic model and analysis of a 2-dof RR spatial robot

clear all, clc

disp('*** Dynamic model and analysis of a 2-dof RR spatial robot ***')
disp(' ')

disp('kinematic quantities, possibly used later in dynamics')
```

```

syms alpha d a theta real
syms q1 q2 L1 L2 real

% number of joints
N=2;

% DH table of parameters
DHTABLE = [ -pi/2      0      sym('L1')  sym('q1');
            0      sym('L2')      0      sym('q2')];

% DH transformation matrix
TDH = [ cos(theta) -sin(theta)*cos(alpha)  sin(theta)*sin(alpha)  a*cos(theta);
        sin(theta)  cos(theta)*cos(alpha) -cos(theta)*sin(alpha)  a*sin(theta);
        0          sin(alpha)           cos(alpha)           d;
        0          0                   0                   1];

A = cell(1,N);
for i = 1:N
alpha = DHTABLE(i,1);
a = DHTABLE(i,2);
d = DHTABLE(i,3);
theta = DHTABLE(i,4);
Ai = subs(TDH);
end

A01=A1;
R01=A01(1:3,1:3)
p01=A01(1:3,4)
r01=simplify(R01'*p01)
A12=A2;
R12=A12(1:3,1:3)
p12=A12(1:3,4)
r12=simplify(R12'*p12)

disp('end-effector position and associated Jacobian (in base frame)')

p_ee_hom=A01*A12*[0 0 0 1]';
p_ee=p_ee_hom(1:3)

q=[q1 q2]';
J_ee=jacobian(p_ee,q)

disp('dynamic parameters')

syms dq1 dq2 ddq1 ddq2 g0 real
syms m1 m2 rc1x rc1y rc1z rc2x real
syms Ic1xx Ic1xy Ic1xz Ic1yy Ic1yz Ic1zz real
syms Ic2xx Ic2yy Ic2zz real
syms a1 a2 a3 a4 a5 a6 real

rc1=[rc1x rc1y rc1z]'
rc2=[rc2x 0 0]'

Ic1=[Ic1xx Ic1xy Ic1xz;
      Ic1xy Ic1yy Ic1yz;
      Ic1xz Ic1yz Ic1zz]
Ic2=diag([Ic2xx Ic2yy Ic2zz])

```

```

gv0=[-g0 0 0]';
disp('moving frames algorithm for computing link kinetic energies')

om0=zeros(3,1);
v0=zeros(3,1);
z0=[0 0 1]';

om1=R01'*(om0+dq1*z0)
v1=R01'*v0+cross(om1,r01)
vc1=simplify(v1+cross(om1,rc1))
T1=simplify(0.5*m1*norm(vc1)^2+0.5*om1'*Ic1*om1);
% ADL: extra step!
T1=collect(T1,m1)

om2=R12'*(om1+dq2*z0)
v2=R12'*v1+cross(om2,r12)
vc2=simplify(v2+cross(om2,rc2))
T2=simplify(0.5*m2*norm(vc2)^2+0.5*om2'*Ic2*om2);
% ADL: extra steps!
T2=subs(T2,sin(q2)^2,1-cos(q2)^2);
T2=collect(T2,cos(q2))

disp('robot kinetic energy')

T=simplify(T1+T2)

disp('robot inertia matrix')

dq=[dq1 dq2]';
M=simplify(hessian(T,dq))

disp('linear parametrization of inertia matrix')

% ADL: by inspection!
% a1=Ic1yy+m1*(rc1x^2+rc1z^2)+Ic2xx
% a2=Ic2yy-Ic2xx+m2*(L2+rc2x)^2
% a3=Ic2zz+m2*(L2+rc2x)^2

M=subs(M,Ic1yy+m1*(rc1x^2+rc1z^2)+Ic2xx,Ic2yy-Ic2xx+m2*(L2+rc2x)^2,Ic2zz+m2*(L2+rc2x)^2,a1,a2,a3)

disp('Christoffel matrices')

M1=M(:,1);
C1=(1/2)*(jacobian(M1,q)+jacobian(M1,q)'+diff(M,q1))
M2=M(:,2);
C2=(1/2)*(jacobian(M2,q)+jacobian(M2,q)'+diff(M,q2))

disp('robot centrifugal and Coriolis terms')

c1=dq'*C1*dq
c2=dq'*C2*dq
c=[c1;c2]

disp('skew-symmetric factorization of velocity terms')

S1=dq'*C1;
S2=dq'*C2;
S=[S1;S2]

```



```

disp('check skew-symmetry of N=dM-2*S')
dM=diff(M,q2)*dq2 % ADL: +diff(M,q1)*dq1 is useless here ...
N=simplify(dM-2*S)
disp('computing link potential energies')
rc01_hom=A01*[rc1;1];
rc01=rc01_hom(1:3)
U1=-m1*gv0'*rc01
rc02_hom=A01*A12*[rc2;1];
rc02=rc02_hom(1:3)
U2=-m2*gv0'*rc02;
% ADL: extra steps!
U2=collect(U2,cos(q2));
U2temp=collect(U2/cos(q2),cos(q1));
U2=U2temp*cos(q2)
disp('robot potential energy')
U=simplify(U1+U2)
disp('gravity terms')
g=jacobian(U,q)
disp('linear parametrization of gravity terms')
% ADL: by inspection! (g0 is known)
% a4=-m2*(L2+rc2x)
% a5=-m1*rc1x
% a6=-m1*rc1z
% ADL: proceed separately!
g=subs(g,-m2*(L2+rc2x),a4);
% ADL: extra step!
g=expand(g);
g=subs(g,-m1*rc1x,-m1*rc1z,a5,a6)
disp(['regressor matrix Y and vector a of dynamic coefficients ' ...
'in a linear parametrization Y(q,dq,ddq)*a=tau'])
ddq=[ddq1 ddq2]';
tau=M*ddq+c+g;
a_symb=[a1 a2 a3 a4 a5 a6]';
Y=simplify(jacobian(tau,a_symb))
a=subs(a_symb,a1,a2,a3,a4,a5,a6,Ic1yy+m1*(rc1x^2+rc1z^2)+Ic2xx,...
Ic2yy-Ic2xx+m2*(L2+rc2x)^2,Ic2zz+m2*(L2+rc2x)^2,-m2*(L2+rc2x),-m1*rc1x,-m1*rc1z)
disp('inverse dynamics computation for q1d=2t, q2d=pi/2')
syms t real
q1d=2*t;
q2d=pi/4;
dq1d=2;
dq2d=0;

```

```

ddq1d=0;
ddq2d=0;

taud=subs(tau,q1,q2,dq1,dq2,ddq1,ddq2,q1d,q2d,dq1d,dq2d,ddq1d,ddq2d);
% ADL: extra step!
taud=collect(taud,sin(2*t))

disp('search of equilibrium configurations qe: g(qe)=0')

% cos(q1)=0 --> g(2)=0 % ADL: requires that a5<a4

q1_e1=pi/2
g_q1_0=subs(g,q1,q1_e1);q2_e1=solve(g_q1_0(1)==0,q2,'Real',true)

% with q1_e1=pi/2
% q2_e1=pi - acos(a5/a4)
% q2_e1=pi + acos(a5/a4)
% ADL: with q1_e1=-pi/2, it is the same pair!

% sin(q2)=0 --> g(2)=0 % ADL: always valid!

q2_e1=0
g_q2_0=subs(g,q2,q2_e1);q1_e1=solve(g_q2_0(1)==0,q1,'Real',true)

% with q2_e1=0
% q1_e1=2*atan((a4 + a5 + (a4^2 + 2*a4*a5 + a5^2 + a6^2)^(1/2))/a6)
% q1_e1=2*atan((a4 + a5 - (a4^2 + 2*a4*a5 + a5^2 + a6^2)^(1/2))/a6)
% ADL: with q2_e1=pi (not shown)
% q1_e2=-2*atan((a4 - a5 + (a4^2 - 2*a4*a5 + a5^2 + a6^2)^(1/2))/a6)
% q1_e2= 2*atan((a5 - a4 + (a4^2 - 2*a4*a5 + a5^2 + a6^2)^(1/2))/a6)

disp('mechanical conditions for balancing gravity: g(q)=0, for all q')

rc2x=-L2 %a4=0
rc1x=0 %a5=0
rc1z=0 %a6=0
a_0=eval(subs(a,rc1x,rc1z,rc2x,0,0,-L2))
g_0=eval(subs(g,a4,a5,a6,a_0(4),a_0(5),a_0(6)))

disp(['tip acceleration ddp for a tip force F in horizontal plane (y0,z0), ' ...
'when the robot is at rest (dq=0) and tau=g=0'])

syms Fy Fz real

J_ee_yz=J_ee(2:3,:)
M_ee_inv=J_ee_yz*M*J_ee_yz'
F=[Fy Fz]';
ddp=simplify(M_ee_inv*F)

% end

```

Exercise #3

This is quite simple. When the robot end-effector is in free space, the backward recursion of the Newton-Euler algorithm is initialized from the last link with zero forces and moments. The gravity term is removed from the picture (the subscript $\alpha = 0$ implies $g_0 = 0$, and thus $\mathbf{g}(\mathbf{q}) = \mathbf{0}$), while the velocity and the inertial terms are zero because $\dot{\mathbf{q}} = \mathbf{0}$ and $\ddot{\mathbf{q}} = \mathbf{0}$. Therefore, one has

$$\mathbf{u}^{(i)} = NE_0(\mathbf{q}, \mathbf{0}, \mathbf{0}) = \mathbf{0}.$$

On the other hand, when a force $\mathbf{f}_e \in \mathbb{R}^3$ and/or a moment $\boldsymbol{\tau}_e \in \mathbb{R}^3$ is applied to the robot end-effector, the components of this wrench are used to initialize the backward recursion from the last link, so that one obtains in this case

$$\mathbf{u}^{(ii)} = NE_0(\mathbf{q}, \mathbf{0}, \mathbf{0}) = \mathbf{J}^T(\mathbf{q})\mathbf{F}_e, \quad \text{with } \mathbf{F}_e = \begin{pmatrix} \mathbf{f}_e \\ \boldsymbol{\tau}_e \end{pmatrix},$$

where $\mathbf{J}(\mathbf{q})$ is the 6×6 geometric Jacobian of the robot.

Exercise #4

In the absence of dissipative effects and assuming that no external contact forces/moments are present, the time evolution of the total energy $E = T(\mathbf{q}, \dot{\mathbf{q}}) + U(\mathbf{q})$ is given by

$$\dot{E} = \dot{\mathbf{q}}^T \boldsymbol{\tau},$$

where $\boldsymbol{\tau} \in \mathbb{R}^n$ is the joint torque delivered to the robot by the motors. When $\dot{\mathbf{q}} \neq \mathbf{0}$, the maximum instantaneous decrease of E under the given torque bounds is obtained by choosing a command $\boldsymbol{\tau}_0^{(i)}$ with components given by

$$\tau_{0,j}^{(i)} = \begin{cases} -T_j \operatorname{sign} \dot{q}_j, & \text{if } \dot{q}_j \neq 0 \\ \text{any}, & \text{if } \dot{q}_j = 0, \end{cases} \quad j = 1, \dots, n.$$

The maximum decrease of E will be

$$\dot{E} = - \sum_{\substack{j=1, \dots, n \\ \dot{q}_j \neq 0}} T_j |\dot{q}_j| < 0.$$

Only the joints that are currently in motion will contribute to the instantaneous reduction of E . For the others, one can simply set $\tau_{0,j}^{(i)} = 0$.

When $\dot{\mathbf{q}} = \mathbf{0}$, then $\dot{E} = 0$ no matter how $\boldsymbol{\tau}$ is chosen. Thus, in order to induce a decrease in the immediate future evolution of E , one should look at the second time derivative of E , namely

$$\ddot{E} = \ddot{\mathbf{q}}^T \boldsymbol{\tau} + \dot{\mathbf{q}}^T \dot{\boldsymbol{\tau}} = \ddot{\mathbf{q}}^T \boldsymbol{\tau},$$

where the second term vanishes being $\dot{\mathbf{q}} = \mathbf{0}$. The joint acceleration $\ddot{\mathbf{q}}$ will depend only on the current robot configuration \mathbf{q}_0 and on the input torque $\boldsymbol{\tau}$, being now $\dot{\mathbf{q}} = \mathbf{0}$ (and thus $\mathbf{c}(\mathbf{q}_0, \mathbf{0}) = \mathbf{0}$):

$$\ddot{\mathbf{q}} = \mathbf{M}^{-1}(\mathbf{q}_0) (\boldsymbol{\tau} - \mathbf{g}(\mathbf{q}_0)).$$

As a result, the second time derivative of E becomes

$$\ddot{E} = \boldsymbol{\tau}^T \mathbf{M}^{-1}(\mathbf{q}_0) \boldsymbol{\tau} - \mathbf{g}^T(\mathbf{q}_0) \mathbf{M}^{-1}(\mathbf{q}_0) \boldsymbol{\tau},$$

namely a complete quadratic function of the input torque $\boldsymbol{\tau}$ to be chosen, with a positive definite quadratic term ($\boldsymbol{\tau}^T \mathbf{M}^{-1}(\mathbf{q}_0) \boldsymbol{\tau} > 0$, for all $\boldsymbol{\tau} \neq \mathbf{0}$, from the property of the robot inertia matrix and of its inverse). The maximum decrease of E in the immediate future will be obtained by choosing $\boldsymbol{\tau}$ so as to minimize $\ddot{E} < 0$ (i.e., imposing the largest possible absolute value of \ddot{E} , with a leading minus sign). To find stationary points for the function $\ddot{E}(\boldsymbol{\tau})$, we set

$$\nabla_{\boldsymbol{\tau}} \ddot{E} = \left(\frac{\partial \ddot{E}}{\partial \boldsymbol{\tau}} \right)^T = 2\mathbf{M}^{-1}(\mathbf{q}_0) \boldsymbol{\tau} - \mathbf{M}^{-1}(\mathbf{q}_0) \mathbf{g}(\mathbf{q}_0) = \mathbf{0}.$$

This linear equation has the unique solution

$$\boldsymbol{\tau}_0^{(ii)} = \frac{\mathbf{g}(\mathbf{q}_0)}{2} \neq \mathbf{0},$$

having assumed that \mathbf{q}_0 is not an (unforced) equilibrium configuration for the robot. Being the Hessian matrix $\nabla_{\boldsymbol{\tau}}^2 \ddot{E} = 2\mathbf{M}^{-1}(\mathbf{q}_0) > 0$, $\boldsymbol{\tau}_0^{(ii)}$ is a minimum for \ddot{E} with respect to $\boldsymbol{\tau}$. Moreover, this torque is certainly feasible for the given bounds — by at least one order of magnitude! The attained value of \ddot{E} is

$$\ddot{E}(\boldsymbol{\tau}_0^{(ii)}) = -\frac{1}{4} \mathbf{g}^T(\mathbf{q}_0) \mathbf{M}^{-1}(\mathbf{q}_0) \mathbf{g}(\mathbf{q}_0) < 0.$$

It is easy to verify that the following expressions hold for the individual contributions to the total energy (kinetic and potential energy) at the given instant of time t_0 , when $\mathbf{x}(t_0) = (\mathbf{q}_0, \mathbf{0})$ and $\boldsymbol{\tau}(t_0) = \boldsymbol{\tau}_0^{(ii)}$:

$$T = 0, \quad \dot{T} = 0, \quad \ddot{T} = \frac{1}{4} \mathbf{g}^T(\mathbf{q}_0) \mathbf{M}^{-1}(\mathbf{q}_0) \mathbf{g}(\mathbf{q}_0) > 0,$$

and

$$U = U(\mathbf{q}_0) \geq 0, \quad \dot{U} = 0, \quad \ddot{U} = -\frac{1}{2} \mathbf{g}^T(\mathbf{q}_0) \mathbf{M}^{-1}(\mathbf{q}_0) \mathbf{g}(\mathbf{q}_0) < 0.$$

Note finally that if we had $\mathbf{g}(\mathbf{q}_0) = \mathbf{0}$, then also $\ddot{E} = 0$ and two situations may arise. Either $U(\mathbf{q}_0) > 0$, and one should then investigate the sign of higher order time derivatives (\ddot{E} and more) in order to assess the evolution of the total energy; or $U(\mathbf{q}_0) = 0$, and then the robot is at $E(t_0) = 0$, a global minimum of the total energy, which cannot be decreased in any case.

* * * * *

*** Projected Gradient and Task Priority (TP) methods for a 3R planar robot ***

kinematics of the 3R planar robot (links of unitary length)

pe_simb =

$\cos(q_1 + q_2 + q_3) + \cos(q_1 + q_2) + \cos(q_1)$
 $\sin(q_1 + q_2 + q_3) + \sin(q_1 + q_2) + \sin(q_1)$

pm_simb =

$\cos(q_1 + q_2) + \cos(q_1)$
 $\sin(q_1 + q_2) + \sin(q_1)$

Je_simb =

$[-\sin(q_1 + q_2 + q_3) - \sin(q_1 + q_2) - \sin(q_1), -\sin(q_1 + q_2 + q_3) - \sin(q_1 + q_2), -\sin(q_1 + q_2 + q_3)]$
 $[\cos(q_1 + q_2 + q_3) + \cos(q_1 + q_2) + \cos(q_1), \cos(q_1 + q_2 + q_3) + \cos(q_1 + q_2), \cos(q_1 + q_2 + q_3)]$

Jm_simb =

$[-\sin(q_1 + q_2) - \sin(q_1), -\sin(q_1 + q_2), 0]$
 $[\cos(q_1 + q_2) + \cos(q_1), \cos(q_1 + q_2), 0]$

at the given configuration (as in the text figure)

q0 =

0
1.5708
-1.5708

numerical kinematics

pe =

2
1

pm =

1
1

Je =

$[-1, -1, 0]$
 $[2, 1, 1]$

Jm =

$[-1, -1, 0]$
 $[1, 0, 0]$

clearance

H =

$$2^{1/2} - 1/2$$

desired end-effector velocity

ve =

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

projected gradient (PG) method

nablaH =

$$\begin{bmatrix} -2^{1/2}/2 \\ -2^{1/2}/4 \\ 0 \end{bmatrix}$$

nablaH =

$$\begin{bmatrix} -0.7071 \\ -0.3536 \\ 0 \end{bmatrix}$$

alfa1 =

$$1$$

Je_pinv =

$$\begin{bmatrix} 0, & 1/3 \\ -1, & -1/3 \\ 1, & 2/3 \end{bmatrix}$$

dq_PG =

$$\begin{bmatrix} 1/3 - 2^{1/2}/12 \\ 2^{1/2}/12 - 1/3 \\ 2^{1/2}/12 + 2/3 \end{bmatrix}$$

dq_PG =

$$\begin{bmatrix} 0.2155 \\ -0.2155 \\ 0.7845 \end{bmatrix}$$

check PG solution

ve_check =

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

vm =

$$\begin{bmatrix} 0 \\ 0.2155 \end{bmatrix}$$

Task Priority (TP) - case A

Pe =

[1/3, -1/3, -1/3]

[-1/3, 1/3, 1/3]

[-1/3, 1/3, 1/3]

vm_A =

0

0.2155

dq_TP_A =

3881782278985403/18014398509481984

-3881782278985403/18014398509481984

14132616230496581/18014398509481984

dq_TP_A =

0.2155

-0.2155

0.7845

check TP_A solution

ve_check =

0

1

vm_check =

0

0.2155

Task Priority (TP) - case B

alfa2 =

1

vm_B =

$1 - 2^{1/2}/4$

$2^{1/2}/4 - 1$

vm_B =

0.6464

-0.6464

dq_TP_B =

-2911336709239053/4503599627370496

2911336709239053/4503599627370496
7414936336609549/4503599627370496

dq_TP_B =

-0.6464
0.6464
1.6464

check TP_B solution

ve_check =

0
1

vm_check =

0
-0.6464

vm_check_error =

0.6464
0

>>

*** Dynamic model and analysis of a 2-dof RR spatial robot ***

kinematic quantities, possibly used later in dynamics

R01 =

$$\begin{bmatrix} \cos(q_1) & 0 & -\sin(q_1) \\ \sin(q_1) & 0 & \cos(q_1) \\ 0 & -1 & 0 \end{bmatrix}$$

p01 =

$$\begin{bmatrix} 0 \\ 0 \\ L_1 \end{bmatrix}$$

r01 =

$$\begin{bmatrix} 0 \\ -L_1 \\ 0 \end{bmatrix}$$

R12 =

$$\begin{bmatrix} \cos(q_2) & -\sin(q_2) & 0 \\ \sin(q_2) & \cos(q_2) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

p12 =

$$\begin{bmatrix} L_2 \cos(q_2) \\ L_2 \sin(q_2) \\ 0 \end{bmatrix}$$

r12 =

$$\begin{bmatrix} L_2 \\ 0 \\ 0 \end{bmatrix}$$

end-effector position and associated Jacobian (in base frame)

p_ee =

$$\begin{bmatrix} L_2 \cos(q_1) \cos(q_2) \\ L_2 \cos(q_1) \sin(q_2) \\ L_1 - L_2 \sin(q_1) \end{bmatrix}$$

J_ee =

$$\begin{bmatrix} -L_2 \cos(q_2) \sin(q_1) & -L_2 \cos(q_1) \sin(q_2) \\ L_2 \cos(q_1) \cos(q_2) & -L_2 \sin(q_1) \sin(q_2) \\ 0 & -L_2 \cos(q_2) \end{bmatrix}$$

dynamic parameters

rc1 =

rc1x
rc1y
rc1z

rc2 =

rc2x
0
0

lc1 =

[lc1xx, lc1xy, lc1xz]
[lc1xy, lc1yy, lc1yz]
[lc1xz, lc1yz, lc1zz]

lc2 =

[lc2xx, 0, 0]
[0, lc2yy, 0]
[0, 0, lc2zz]

gv0 =

-g0
0
0

moving frames algorithm for computing link kinetic energies

om1 =

0
-dq1
0

v1 =

0
0
0

vc1 =

-dq1*rc1z
0
dq1*rc1x

T1 =

$((dq1^2*(rc1x^2 + rc1z^2))/2)*m1 + (lc1yy*dq1^2)/2$

om2 =

-dq1*sin(q2)
-dq1*cos(q2)

dq2

v2 =

$$\begin{bmatrix} 0 \\ L2 \cdot dq2 \\ L2 \cdot dq1 \cdot \cos(q2) \end{bmatrix}$$

vc2 =

$$\begin{bmatrix} 0 \\ dq2 \cdot (L2 + rc2x) \\ dq1 \cdot \cos(q2) \cdot (L2 + rc2x) \end{bmatrix}$$

T2 =

$$\frac{(Ic2yy \cdot dq1^2)/2 - (Ic2xx \cdot dq1^2)/2 + (dq1^2 \cdot m2 \cdot (L2 + rc2x)^2)/2}{2} \cdot \cos(q2)^2 + \frac{(Ic2xx \cdot dq1^2)/2 + (Ic2zz \cdot dq2^2)/2 + (dq2^2 \cdot m2 \cdot (L2 + rc2x)^2)/2}{2}$$

robot kinetic energy

T =

$$\frac{(Ic1yy \cdot dq1^2)/2 + ((Ic2yy \cdot dq1^2)/2 - (Ic2xx \cdot dq1^2)/2 + (dq1^2 \cdot m2 \cdot (L2 + rc2x)^2)/2) \cdot \cos(q2)^2 + ((dq1^2 \cdot (rc1x^2 + rc1z^2))/2) \cdot m1 + (Ic2xx \cdot dq1^2)/2 + (Ic2zz \cdot dq2^2)/2 + (dq2^2 \cdot m2 \cdot (L2 + rc2x)^2)/2}{2}$$

robot inertia matrix

M =

$$\begin{bmatrix} Ic2xx + Ic1yy + \cos(q2)^2 \cdot (Ic2yy - Ic2xx + m2 \cdot (L2 + rc2x)^2) + m1 \cdot (rc1x^2 + rc1z^2), & 0 \\ 0, & Ic2zz + m2 \cdot (L2 + rc2x)^2 \end{bmatrix}$$

linear parametrization of inertia matrix

M =

$$\begin{bmatrix} a2 \cdot \cos(q2)^2 + a1, & 0 \\ 0, & a3 \end{bmatrix}$$

Christoffel matrices

C1 =

$$\begin{bmatrix} 0, & -a2 \cdot \cos(q2) \cdot \sin(q2) \\ -a2 \cdot \cos(q2) \cdot \sin(q2), & 0 \end{bmatrix}$$

C2 =

$$\begin{bmatrix} a2 \cdot \cos(q2) \cdot \sin(q2), & 0 \\ 0, & 0 \end{bmatrix}$$

robot centrifugal and Coriolis terms

c1 =

$$-2 \cdot a2 \cdot dq1 \cdot dq2 \cdot \cos(q2) \cdot \sin(q2)$$

c2 =

$$a2*dq1^2*cos(q2)*sin(q2)$$

c =

$$\begin{aligned} & -2*a2*dq1*dq2*cos(q2)*sin(q2) \\ & a2*dq1^2*cos(q2)*sin(q2) \end{aligned}$$

skew-symmetric factorization of velocity terms

S =

$$\begin{bmatrix} -a2*dq2*cos(q2)*sin(q2), & -a2*dq1*cos(q2)*sin(q2) \\ a2*dq1*cos(q2)*sin(q2), & 0 \end{bmatrix}$$

check skew-symmetry of $N=dM-2*S$

dM =

$$\begin{bmatrix} -2*a2*dq2*cos(q2)*sin(q2), & 0 \\ 0, & 0 \end{bmatrix}$$

N =

$$\begin{bmatrix} 0, & a2*dq1*sin(2*q2) \\ -a2*dq1*sin(2*q2), & 0 \end{bmatrix}$$

computing link potential energies

rc01 =

$$\begin{aligned} & rc1x*cos(q1) - rc1z*sin(q1) \\ & rc1z*cos(q1) + rc1x*sin(q1) \\ & L1 - rc1y \end{aligned}$$

U1 =

$$g0*m1*(rc1x*cos(q1) - rc1z*sin(q1))$$

rc02 =

$$\begin{aligned} & L2*cos(q1)*cos(q2) + rc2x*cos(q1)*cos(q2) \\ & L2*cos(q2)*sin(q1) + rc2x*cos(q2)*sin(q1) \\ & L1 - L2*sin(q2) - rc2x*sin(q2) \end{aligned}$$

U2 =

$$cos(q1)*cos(q2)*g0*m2*(L2 + rc2x)$$

robot potential energy

U =

$$g0*m1*(rc1x*cos(q1) - rc1z*sin(q1)) + g0*m2*cos(q1)*cos(q2)*(L2 + rc2x)$$

gravity terms

g =

$$-g_0*m_1*(rc_{1z}*\cos(q_1) + rc_{1x}*\sin(q_1)) - g_0*m_2*\cos(q_2)*\sin(q_1)*(L_2 + rc_{2x}) \\ -g_0*m_2*\cos(q_1)*\sin(q_2)*(L_2 + rc_{2x})$$

linear parametrization of gravity terms

g =

$$a_5*g_0*\sin(q_1) + a_6*g_0*\cos(q_1) + a_4*g_0*\cos(q_2)*\sin(q_1) \\ a_4*g_0*\cos(q_1)*\sin(q_2)$$

regressor matrix Y and vector a of dynamic coefficients in a linear parametrization $Y(q,dq,ddq)*a=\tau$

Y =

$$[ddq_1, ddq_1*\cos(q_2)^2 - 2*dq_1*dq_2*\cos(q_2)*\sin(q_2), 0, g_0*\cos(q_2)*\sin(q_1), g_0*\sin(q_1), g_0*\cos(q_1)] \\ [0, (dq_1^2*\sin(2*q_2))/2, ddq_2, g_0*\cos(q_1)*\sin(q_2), 0, 0]$$

a =

$$lc_{2xx} + lc_{1yy} + m_1*(rc_{1x}^2 + rc_{1z}^2) \\ lc_{2yy} - lc_{2xx} + m_2*(L_2 + rc_{2x})^2 \\ lc_{2zz} + m_2*(L_2 + rc_{2x})^2 \\ - m_2*(L_2 + rc_{2x}) \\ - m_1*rc_{1x} \\ - m_1*rc_{1z}$$

inverse dynamics computation for $q_1d=2t, q_2d=\pi/2$

taud =

$$(a_5*g_0 + (2^{1/2}*a_4*g_0)/2)*\sin(2*t) + a_6*g_0*\cos(2*t) \\ 2*a_2 + (2^{1/2}*a_4*g_0*\cos(2*t))/2$$

search of equilibrium configurations $q_e: g(q_e)=0$

q1_e1 =

1.5708

Warning: Solutions are only valid under certain conditions. To include parameters and conditions in the solution, specify the 'ReturnConditions' value as 'true'.

> In sym/solve>warnIfParams (line 478)

In sym/solve (line 357)

In Ex2_Midterm (line 189)

q2_e1 =

pi - acos(a5/a4)

pi + acos(a5/a4)

q2_e1 =

0

q1_e1 =

$$2*\text{atan}((a4 + a5 + (a4^2 + 2*a4*a5 + a5^2 + a6^2)^{1/2})/a6)$$
$$2*\text{atan}((a4 + a5 - (a4^2 + 2*a4*a5 + a5^2 + a6^2)^{1/2})/a6)$$

mechanical conditions for balancing gravity: $g(q)=0$, for all q

rc2x =

-L2

rc1x =

0

rc1z =

0

a_0 =

lc2xx + lc1yy

lc2yy - lc2xx

lc2zz

0

0

0

g_0 =

0

0

tip acceleration ddp for a tip force F in horizontal plane (y_0, z_0) , when the robot is at rest ($dq=0$) and $\tau=g=0$

J_ee_yz =

$$[L2*\cos(q1)*\cos(q2), -L2*\sin(q1)*\sin(q2)]$$

$$[\quad 0, \quad -L2*\cos(q2)]$$

M_ee_inv =

$$[L2^2*\cos(q1)^2*\cos(q2)^2*(a2*\cos(q2)^2 + a1) + L2^2*a3*\sin(q1)^2*\sin(q2)^2,$$

$$L2^2*a3*\cos(q2)*\sin(q1)*\sin(q2)]$$

$$[\quad \quad \quad L2^2*a3*\cos(q2)*\sin(q1)*\sin(q2), \quad \quad L2^2*a3*\cos(q2)^2]$$

ddp =

$$Fy*(L2^2*\cos(q1)^2*\cos(q2)^2*(a2*\cos(q2)^2 + a1) + L2^2*a3*\sin(q1)^2*\sin(q2)^2) +$$

$$Fz*L2^2*a3*\cos(q2)*\sin(q1)*\sin(q2)$$

$$L2^2*a3*\cos(q2)*(Fz*\cos(q2) + Fy*\sin(q1)*\sin(q2))$$

>>