

ROB510 Robot Kinematics and Dynamics

Homework 4

Jackson Chen

Friday 29th March, 2024

U M I C H
R O B O T
T E N E T
T O B O R
H C I M U

Problem I

For manipulator (i) in Figure 3.23 (MLS book), find the spatial Jacobian. Please find this by hand; do not use Mathematica or the equivalent. Use a spatial frame with its origin at the intersection of ξ_1 and ξ_2 and principal axes parallel to those used in the book; (i.e., z vertical, y to the right, x out of the page).

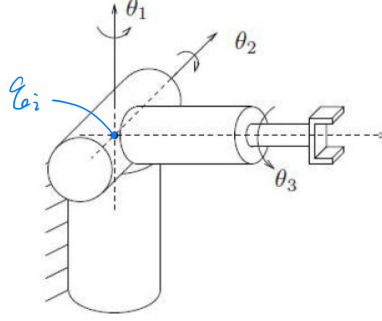


Figure 1: Problem 1

According to the example given in lecture, we have

$$J_{st}^s = [\xi'_1 \quad \xi'_2 \quad \xi'_3],$$

where

$$\begin{aligned} \xi'_1 &= \begin{bmatrix} -\omega_1 \times q'_1 \\ \omega_1 \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \\ \xi'_2 &= \begin{bmatrix} -\omega_2 \times q'_2 \\ \omega_2 \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} \cos(\theta_1) \\ \sin(\theta_1) \\ 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \\ -\cos(\theta_1) \\ -\sin(\theta_1) \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -\cos(\theta_1) \\ -\sin(\theta_1) \\ 0 \end{bmatrix}, \\ \xi'_3 &= \begin{bmatrix} -\omega_3 \times q'_3 \\ \omega_3 \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} \sin(\theta_1)\cos(\theta_2) \\ -\cos(\theta_1)\cos(\theta_2) \\ \sin(\theta_2) \\ -\sin(\theta_1)\cos(\theta_2) \\ \cos(\theta_1)\cos(\theta_2) \\ -\sin(\theta_2) \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \\ -\sin(\theta_1)\cos(\theta_2) \\ \cos(\theta_1)\cos(\theta_2) \\ -\sin(\theta_2) \end{bmatrix}. \end{bmatrix}$$

Note that the new ω_2 and ω_3 are calculated by multiplying the original rotation axis by the first and first & second rotations. The rotation matrices for the first and second rotations can be calculated using the Rodrigues' Equation.

Problem II

Suppose that each of the four manipulators in Figure 3.24 of MLS experiences the following wrench on their end-effectors (Given in body coordinates): $\mathbf{F}_B = [f_x, f_y, f_z, \tau_x, \tau_y, \tau_z]^T$. Use the values of $\mathbf{q}, \omega, \mathbf{g}_s(0)$ and the resulting twists ξ as shown in the solutions of homework 2, problem 3, and $\theta = [0, \pi/2, 0, 0, 0, 0]^T$ for each manipulator.

Note: Problem 2 allows the use of Mathematica. You are also permitted to use the "Screws" and "RobotLinks" packages, but specifically not the BodyJacobian and SpatialJacobian functions (or any other function that finds the Jacobian directly). Copy and paste your code in the homework.

- Find the joint torque array τ that would be necessary to resist the wrench \mathbf{F}_B for each manipulator. Note: in this context, "resist the wrench" means to exert the appropriate joint torque to keep the manipulator in equilibrium, i.e., sum of forces = 0.
- If not all of six terms of \mathbf{F}_B appear in the solution of τ , explain why (your explanation should be general, not for each manipulator individually).

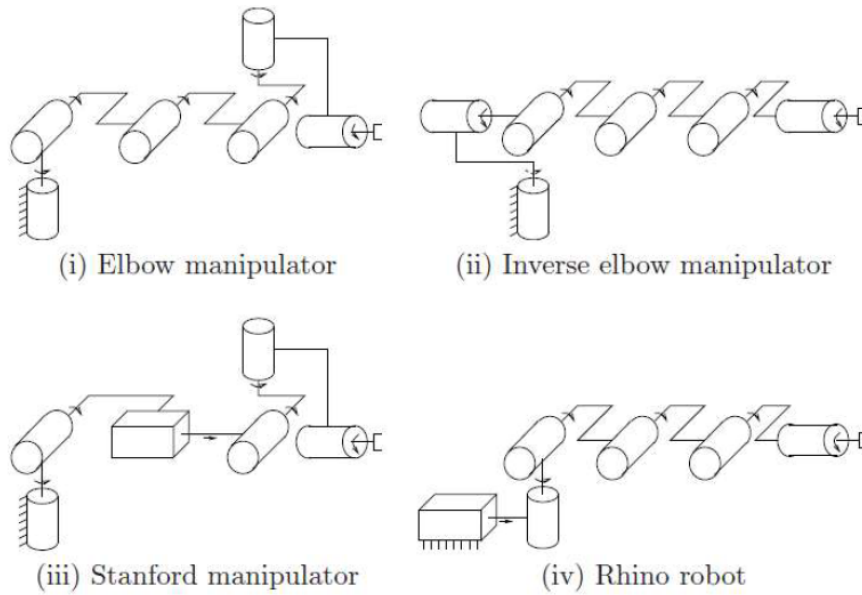


Figure 2: Four Different Types of Manipulators

```

1 (* Mathematica Code *)
2 ROB510 Homework4
3 Needs["Screws", \
4 "C:\Users\Alex\OneDrive\          \OneDrive\ROB510\Homework\Mathematica\S\
5 crews.m"]
6 Elbow Manipulator
7 g0 = {{1, 0, 0, 0}, {0, 1, 0, 11 + 12 + 13}, {0, 0, 1, 0}, {0, 0, 0,
8      1}};
9 s1 = {0, 0, 0, 0, 0, 1};
10 s2 = {0, 0, 0, \[Minus]1, 0, 0};

```

```

11 s3 = {0, 0, l1, \[Minus]1, 0, 0};
12 s4 = {0, 0, l1 + l2, \[Minus]1, 0, 0};
13 s5 = {l1 + l2, 0, 0, 0, 0, 1};
14 s6 = {0, 0, 0, 0, 1, 0};
15 q1 = {0, 0, 0};
16 q2 = q1;
17 q3 = q2;
18 q4 = {0, l1, 0};
19 q5 = {0, l1 + l2, 0};
20 q6 = q5;
21 e1 = TwistExp[s1, q1];
22 e2 = TwistExp[s2, q2];
23 e3 = TwistExp[s3, q3];
24 e4 = TwistExp[s4, q4];
25 e5 = TwistExp[s5, q5];
26 e6 = TwistExp[s6, q6];
27 MatrixForm[gst = Simplify[e1 . e2 . e3 . e4 . e5 . e6 . g0]];
28 g1 = gst;
29 g2 = e2 . e3 . e4 . e5 . e6 . g0;
30 g3 = e3 . e4 . e5 . e6 . g0;
31 g4 = e4 . e5 . e6 . g0;
32 g5 = e5 . e6 . g0;
33 g6 = e6 . g0;
34 adj1 = Inverse[RigidAdjoint[g1]];
35 adj2 = Inverse[RigidAdjoint[g2]];
36 adj3 = Inverse[RigidAdjoint[g3]];
37 adj4 = Inverse[RigidAdjoint[g4]];
38 adj5 = Inverse[RigidAdjoint[g5]];
39 adj6 = Inverse[RigidAdjoint[g6]];
40 s1' = adj1 . s1;
41 s2' = adj1 . s2;
42 s3' = adj1 . s3;
43 {
44   {s4' = adj1 . s4;},
45   {s5' = adj1 . s5;},
46   {s6' = adj1 . s6;},
47   jacob = -Transpose[{s1', s2', s3', s4', s5', s6}];
48   force = {fx, fy, fz, tx, ty, tz};
49   MatrixForm[Torque = jacob . force]}
50 }
51
52
53 Inverse Elbow Manipulator
54 gst0 = {{1, 0, 0, 0}, {0, 1, 0, l1 + l2 + l3}, {0, 0, 1, 0}, {0, 0, 0,
55   1}};
56 s1 = {0, 0, 0, 0, 0, 1};
57 s2 = {0, 0, 0, -1, 0, 0};
58 s3 = {0, 1, 0, 0, 0, 0};
59 s4 = {0, 0, l1 + l2, \[Minus]1, 0, 0};
60 s5 = {l1 + l2, 0, 0, 0, 0, 1};
61 s6 = {0, 0, 0, 0, 1, 0};
62 q1 = {0, 0, 0};
63 q2 = q1;
64 q3 = {0, l1, 0};
65 q4 = {0, l1 + l2, 0};
66 q5 = q4;
67 q6 = q5;
68 e1 = TwistExp[s1, q1];
69 e2 = TwistExp[s2, q2];
70 e3 = TwistExp[s3, q3];

```

```

71 e4 = TwistExp[s4, q4];
72 e5 = TwistExp[s5, q5];
73 e6 = TwistExp[s6, q6];
74 MatrixForm[gst = Simplify[e1 . e2 . e3 . e4 . e5 . e6 . gst0]];
75 g1 = gst;
76 g2 = e2 . e3 . e4 . e5 . e6 . g0;
77 g3 = e3 . e4 . e5 . e6 . g0;
78 g4 = e4 . e5 . e6 . g0;
79 g5 = e5 . e6 . g0;
80 g6 = e6 . g0;
81 adj1 = Inverse[RigidAdjoint[g1]];
82 adj2 = Inverse[RigidAdjoint[g2]];
83 adj3 = Inverse[RigidAdjoint[g3]];
84 adj4 = Inverse[RigidAdjoint[g4]];
85 adj5 = Inverse[RigidAdjoint[g5]];
86 adj6 = Inverse[RigidAdjoint[g6]];
87 s1' = adj1 . s1;
88 s2' = adj1 . s2;
89 s3' = adj1 . s3;
90 {
91 {s4' = adj1 . s4;},
92 {s5' = adj1 . s5;
93 s6' = adj1 . s6;
94 jacob = -Transpose[{s1', s2', s3', s4', s5', s6}];
95 force = {fx, fy, fz, tx, ty, tz};
96 MatrixForm[Torque = jacob . force]}
97 }
98
99 Stanford Manipulator
100 g0 = {{1, 0, 0, 0}, {0, 1, 0, l1 + l2 + l3}, {0, 0, 1, 0}, {0, 0, 0,
101 1}};
102 s1 = {0, 0, 0, 0, 0, 1};
103 s2 = {0, 0, 0, \[Minus]1, 0, 0};
104 s3 = {0, 1, 0, 0, 0, 0};
105 s4 = {0, 0, l1 + l2, \[Minus]1, 0, 0};
106 s5 = {l1 + l2, 0, 0, 0, 0, 1};
107 s6 = {0, 0, 0, 0, 1, 0};
108 q1 = {0, 0, 0};
109 q2 = q1;
110 q3 = {0, l1, 0};
111 q4 = {0, l1 + l2, 0};
112 q5 = q4;
113 q6 = q5;
114 e1 = TwistExp[s1, q1];
115 e2 = TwistExp[s2, q2];
116 e3 = TwistExp[s3, q3];
117 e4 = TwistExp[s4, q4];
118 e5 = TwistExp[s5, q5];
119 e6 = TwistExp[s6, q6];
120 MatrixForm[gst = Simplify[e1 . e2 . e3 . e4 . e5 . e6 . g0]];
121 g1 = gst;
122 g2 = e2 . e3 . e4 . e5 . e6 . g0;
123 g3 = e3 . e4 . e5 . e6 . g0;
124 g4 = e4 . e5 . e6 . g0;
125 g5 = e5 . e6 . g0;
126 g6 = e6 . g0;
127 adj1 = Inverse[RigidAdjoint[g1]];
128 adj2 = Inverse[RigidAdjoint[g2]];
129 adj3 = Inverse[RigidAdjoint[g3]];
130 adj4 = Inverse[RigidAdjoint[g4]];

```

```

131 adj5 = Inverse[RigidAdjoint[g5]];
132 adj6 = Inverse[RigidAdjoint[g6]];
133 s1' = adj1 . s1;
134 s2' = adj1 . s2;
135 s3' = adj1 . s3;
136 {
137   {s4' = adj1 . s4;},
138   {s5' = adj1 . s5;
139     s6' = adj1 . s6;
140     jacob = -Transpose[{s1', s2', s3', s4', s5', s6}];
141     force = {fx, fy, fz, tx, ty, tz};
142     MatrixForm[Torque = jacob . force]}
143 }
144
145 Rhino Robot
146 g0 = {{1, 0, 0, 0}, {0, 1, 0, l1 + l2 + l3}, {0, 0, 1, 0}, {0, 0, 0,
147   1}};
148 s1 = {0, 1, 0, 0, 0, 0};
149 s2 = {0, 0, 0, 0, 0, 1};
150 s3 = {0, 0, -1, 0, 0, 0};
151 s4 = {0, 0, l1, \[Minus]1, 0, 0};
152 s5 = {0, 0, l1 + l2, -1, 0, 0};
153 s6 = {0, 0, 0, 0, 1, 0};
154 q1 = {0, 0, 0};
155 q2 = q1;
156 q3 = q2;
157 q4 = {0, l1, 0};
158 q5 = {0, l1 + l2, 0};
159 q6 = q5;
160 e1 = TwistExp[s1, q1];
161 e2 = TwistExp[s2, q2];
162 e3 = TwistExp[s3, q3];
163 e4 = TwistExp[s4, q4];
164 e5 = TwistExp[s5, q5];
165 e6 = TwistExp[s6, q6];
166 MatrixForm[gst = Simplify[e1 . e2 . e3 . e4 . e5 . e6 . g0]];
167 g1 = gst;
168 g2 = e2 . e3 . e4 . e5 . e6 . g0;
169 g3 = e3 . e4 . e5 . e6 . g0;
170 g4 = e4 . e5 . e6 . g0;
171 g5 = e5 . e6 . g0;
172 g6 = e6 . g0;
173 adj1 = Inverse[RigidAdjoint[g1]];
174 adj2 = Inverse[RigidAdjoint[g2]];
175 adj3 = Inverse[RigidAdjoint[g3]];
176 adj4 = Inverse[RigidAdjoint[g4]];
177 adj5 = Inverse[RigidAdjoint[g5]];
178 adj6 = Inverse[RigidAdjoint[g6]];
179 s1' = adj1 . s1;
180 s2' = adj1 . s2;
181 s3' = adj1 . s3;
182 {
183   {s4' = adj1 . s4;},
184   {s5' = adj1 . s5;
185     s6' = adj1 . s6;
186     jacob = -Transpose[{s1', s2', s3', s4', s5', s6}];
187     force = {fx, fy, fz, tx, ty, tz};
188     MatrixForm[Torque = jacob . force]}
189 }

```

My output are similar to this (too long to include in the code). Some force components does not

```

out(qq) = ElbowManipulator
      13 ty Cos[11] Cos[11 + 12] + fx (11 + 12 + 13 Cos[11] Cos[11 + 12])
] Sin[11] + 13 Cos[12]^2 Cos[11 + 12]^2 Sin[11] + 11 Sec[11 + 12] Tan[11] + tx (Sec[11]^2 Sec[11 + 12]^2 (-11 Cos[11] Cos[11 + 12] Sin[11] - 12
[11 + 12]^2 + fz (-11 + (11 Cos[11]^2 Cos[11 + 12]^2 + 12 Cos[11]^2 Cos[11 + 12]^2 + 13 Cos[11]^3 Cos[11 + 12]^2) Sec[11]^2 Sec[11 + 12]^2) + tx (-11 + 12
      - tz + fx Sec[11 + 12] Tan[11] + ty Sec[11 + 12] Tan[12] - fy Tan[11 + 12] - fz Tan[11 + 12] - tx Tan[11 + 12]
      fx - ty

```

Figure 3: Result for Elbow Manipulator

generate torque since the structure of the manipulator determine that it can resist them automatically.

Problem III

Euler angles can be used to represent rotations via the product of exponentials formula. If we think of (α, β, γ) as joint angles of a robot manipulator, then we can find the singularities of a Euler angle parameterization by calculating the Jacobian of the "forward kinematics", where we are concerned only with the rotation portion of the forward kinematics map. Use this point of view to find singularities for the following class of Euler angles: i) ZYZ Sequence, and ii) ZXY sequence.

For this problem, essentially we need to calculate the manipulator Jacobian and see if it could lose rank in some scenarios.

1. ZYZ Sequence:

Similar as the process in Problem 1, we can write out

$$\xi'_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \xi'_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -\sin(\theta_1) \\ \cos(\theta_1) \\ 0 \end{bmatrix}$$

$$\omega''_3 = R_z(\theta_1)e^2e^1\omega_3 = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 \\ \sin(\theta_1) & \cos(\theta_1) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \sin(\theta_2) \\ 0 \\ \cos(\theta_2) \end{bmatrix} = \begin{bmatrix} \cos(\theta_1)\sin(\theta_2) \\ \sin(\theta_1)\sin(\theta_2) \\ \cos(\theta_2) \end{bmatrix}$$

Thus, the manipulator Jacobian for ZYZ Sequence is

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & -\sin(\theta_1) & \cos(\theta_1)\sin(\theta_2) \\ 0 & \cos(\theta_1) & \sin(\theta_1)\sin(\theta_2) \\ 1 & 0 & \cos(\theta_2) \end{bmatrix},$$

which will lose rank when $\cos(\theta_2) = 0$, i.e., when $\theta_2 = 0, \pi$, given $\theta \in [0, 2\pi)$.

2. ZXY Sequence:

Similarly, we have the same expression as in Problem 1 except for the negative sign for the x-axis since it was Z-XY in Problem 1.

$$\xi'_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \xi'_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \cos(\theta_1) \\ \sin(\theta_1) \\ 0 \end{bmatrix}, \xi'_3 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \sin(\theta_1)\cos(\theta_2) \\ -\cos(\theta_1)\cos(\theta_2) \\ \sin(\theta_2) \end{bmatrix}$$

Thus, the manipulator Jacobian for ZXY Sequence is

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \cos(\theta_1) & \sin(\theta_1)\cos(\theta_2) \\ 0 & \sin(\theta_1) & -\cos(\theta_1)\cos(\theta_2) \\ 1 & 0 & \sin(\theta_2) \end{bmatrix},$$

which will lose rank when $\sin(\theta_2) = 0$, i.e., when $\theta_2 = \frac{\pi}{2}, \frac{3\pi}{2}$, given $\theta \in [0, 2\pi)$.

Problem IV

Recall for a particle with kinetic energy, $K = \frac{1}{2}m\dot{x}^2$ the momentum is defined as

$$p = m\dot{x} = \frac{dK}{d\dot{x}}$$

Therefore, for a mechanical system with generalized coordinates q_1, \dots, q_n , we define the generalized momentum p_k as

$$p_k = \frac{\partial L}{\partial \dot{q}_k}$$

where L is the Lagrangian of the system. With $K = \frac{1}{2}\dot{q}^\top D(q)\dot{q}$ and $L = K - V$ prove that

$$\sum_{k=1}^n \dot{q}_k p_k = 2K$$

Proof:

$$\sum_{k=1}^n \dot{q}_k p_k = \dot{q} \cdot p = \dot{q} \cdot \left[\frac{d}{d\dot{x}} \frac{1}{2} \dot{q}^\top D(q) \dot{q} \right] = \dot{q} \cdot [D(q)\dot{q}] = \dot{q}^\top D(q)\dot{q} = 2K$$

Q.E.D

Problem V

There is another formulation of the equations of motion of a mechanical system that is useful, the so-called **Hamiltonian** formulation. Define the Hamiltonian function H by

$$H = \sum_{k=1}^n \dot{q}_k p_k - L$$

1. Show that $H = K + V$.

Proof:

The Lagrangian L is given by $L = K - V$, where K is the kinetic energy and V is the potential energy of the system. From previous problem we know that $p_k = \frac{\partial L}{\partial \dot{q}_k}$ and $K = \frac{1}{2} \dot{q}^\top D(q) \dot{q}$.

We have Hamiltonian

$$H = \sum_{k=1}^n \dot{q}_k p_k - L = \sum_{k=1}^n \dot{q}_k p_k - L = \sum_{k=1}^n \dot{q}_k p_k - (K - V)$$

Utilizing the result from Problem 4, i.e., $\sum_{k=1}^n \dot{q}_k p_k = 2K$, we have

$$H = 2K - (K - V) = K + V$$

Q.E.D

2. Using the Euler-Lagrange equations, derive Hamilton's equations.

$$\begin{aligned} \dot{q}_k &= \frac{\partial H}{\partial p_k} \\ \dot{p}_k &= -\frac{\partial H}{\partial q_k} + \tau_k \end{aligned}$$

where τ_k is the input generalized force.

Proof 1:

Euler-Lagrange equation with external force τ_k is given by

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_k} \right) - \frac{\partial L}{\partial q_k} = \tau_k$$

Since we have

$$H = \sum_{k=1}^n \dot{q}_k p_k - L$$

Then we can write

$$\frac{\partial H}{\partial q_k} = -\frac{\partial L}{\partial q_k}$$

since there is no q_k term in any $\dot{q}_k p_k$.

Now we have

$$\frac{\partial H}{\partial q_k} = -\frac{\partial L}{\partial q_k} = \tau_k - \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_k} \right) = \tau_k - \frac{d}{dt} p_k = \tau_k - \dot{p}_k.$$

Thus,

$$\dot{p}_k = -\frac{\partial H}{\partial q_k} + \tau_k.$$

Q.E.D

Proof 2:

Again, from the definition of the Hamiltonian function, we have

$$\frac{\partial H}{\partial p_k} = \frac{\partial}{\partial p_k} \left(\sum_{i=1}^n \dot{q}_i p_i \right) - \frac{\partial L}{\partial p_k},$$

where

$$\frac{\partial}{\partial p_k} \left(\sum_{i=1}^n \dot{q}_i p_i \right) = \dot{q}_k + \sum_{i=1}^n \frac{\partial \dot{q}_i}{\partial p_k} p_i.$$

and

$$\frac{\partial L}{\partial p_k} = \left(\frac{\partial K}{\partial p_k} - \frac{\partial V}{\partial p_k} \right) = \frac{\partial K}{\partial p_k} = \sum_{i=1}^n \frac{\partial K}{\partial \dot{q}_i} \frac{\partial \dot{q}_i}{\partial p_k} = \sum_{i=1}^n p_i \frac{\partial \dot{q}_i}{\partial p_k}.$$

Thus,

$$\frac{\partial H}{\partial p_k} = \dot{q}_k + \sum_{i=1}^n \frac{\partial \dot{q}_i}{\partial p_k} p_i - \sum_{i=1}^n p_i \frac{\partial \dot{q}_i}{\partial p_k} = \dot{q}_k.$$

Q.E.D

Problem VI

For two-link manipulator of Figure 3 compute Hamiltonian equations in vector form. Note that Hamilton's equations are a system of first order differential equations as opposed to a second order system given by Lagrange's equations.

As derived in SHV, the inertia matrix $D(q)$ is given as $D(q) = \begin{bmatrix} d_{11}(q) & d_{12}(q) \\ d_{12}(q) & d_{22}(q) \end{bmatrix}$. The total potential energy is $V(q) = V_1(q) + V_2(q)$.

Note: You do not have to plug in the actual expressions for $d_{ij}(q)$, $V_1(q)$, or $V_2(q)$, but feel free to use Mathematica to see how it turns out.

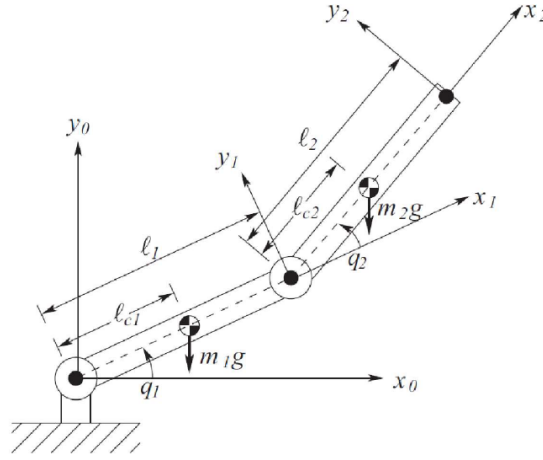


Figure 4: Two-link revolute joint arm. The rotational joint motion introduces dynamic coupling between the joints. (Figure 6.9 in SHV)

We have

$$K = \frac{1}{2} \dot{q}^T D(q) \dot{q}.$$

Since $D(q)$ is defined in such way that

$$p = D(q) \dot{q} \text{ with } p_k = \frac{\partial L}{\partial \dot{q}_k}.$$

since $L = K - V$ and V does not depend on \dot{q} . We could have

$$\dot{q} = D^{-1}(q)p.$$

Now we can write

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} = D^{-1}(q) \begin{bmatrix} p_1 \\ p_2 \end{bmatrix}.$$

Then utilizing Hamiltonian equation, and with $H = K + V$, we have

$$\dot{p}_k = -\frac{\partial H}{\partial q_k} + \tau_k = \begin{bmatrix} \dot{p}_1 \\ \dot{p}_2 \end{bmatrix} = \begin{bmatrix} -\frac{\partial H}{\partial q_1} + \tau_1 \\ -\frac{\partial H}{\partial q_2} + \tau_2 \end{bmatrix} = \begin{bmatrix} -\frac{\partial K(q,p)}{\partial q_1} - \frac{\partial V}{\partial q_1} + \tau_1 \\ -\frac{\partial K(q,p)}{\partial q_2} - \frac{\partial V}{\partial q_2} + \tau_2 \end{bmatrix}.$$