# 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 $\xi_1$ and $\xi_2$ and principal axes parallel to those used in the book; (i.e., z vertical, y to the right, x out of the page).

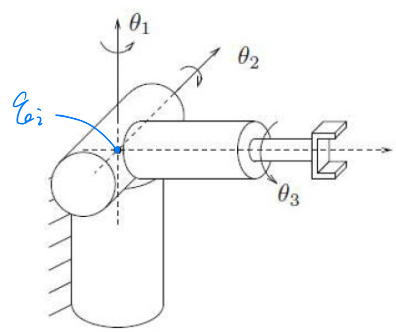


Figure 1: Problem 1

According to the example given in lecture, we have

$$J_{st}^{s} = \begin{bmatrix} \xi_1' & \xi_2' & \xi_3' \end{bmatrix},$$

where

$$\xi_1' = \begin{bmatrix} -\omega_1 \times q_1' \\ \omega_1 \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix},$$

$$\xi_2' = \begin{bmatrix} -\omega_2 \times q_2' \\ \omega_2 \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} cos(\theta_1) \\ sin(\theta_1) \\ 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \\ -cos(\theta_1) \\ -sin(\theta_1) \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -cos(\theta_1) \\ -sin(\theta_1) \\ 0 \end{bmatrix},$$

$$\xi_3' = \begin{bmatrix} -\omega_3 \times q_3' \\ \omega_3 \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} sin(\theta_1)cos(\theta_2) \\ -cos(\theta_1)cos(\theta_2) \\ sin(\theta_2) \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \\ -sin(\theta_1)cos(\theta_2) \\ cos(\theta_1)cos(\theta_2) \\ -sin(\theta_2) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -sin(\theta_1)cos(\theta_2) \\ cos(\theta_1)cos(\theta_2) \\ -sin(\theta_2) \end{bmatrix}.$$

Note that the new $\omega_2$ and $\omega_3$ 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): $\mathrm{FB_B} = [f_x, f_y, f_z, \tau_x, \tau_y, \tau_z]^{\mathrm{T}}$. Use the values of $q, \omega$, gst (0) and the resulting twists $\xi$ as shown in the solutions of homework 2 , problem 3 , and $\theta = [0,\ \pi/2, 0, 0, 0, 0]^{\mathrm{T}}$ 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.

a) Find the joint torque array $\tau$ that would be necessary to resist the wrench $F_B$ 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.

b) If not all of six terms of $\mathrm{F_B}$ appear in the solution of $\tau$, explain why (your explanation should be general, not for each manipulator individually).

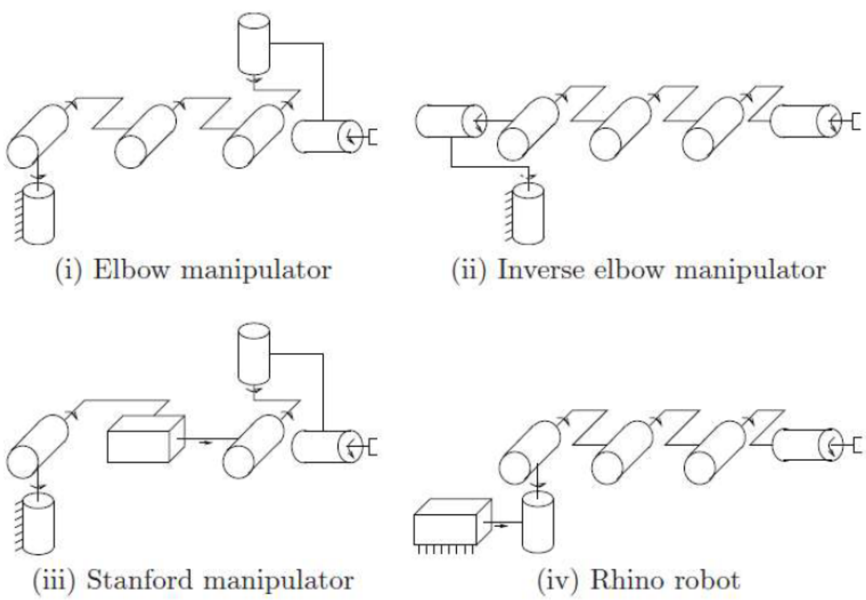


Figure 2: Four Different Types of Manipulators

```
(* Mathematica Code *)
ROB510 Homework4
Needs["Screws`", \
"C:\Users\Alex\OneDrive\          \OneDrive\ROB510\Homework\Mathematica\S\
crews.m"]
Elbow Manipulator
g0 = {{1, 0, 0, 0}, {0, 1, 0, l1 + l2 + l3}, {0, 0, 1, 0}, {0, 0, 0,
    1}};
s1 = {0, 0, 0, 0, 0, 1};
s2 = {0, 0, 0, \[Minus]1, 0, 0};
```

```
s3 = {0, 0, l1, \[Minus]1, 0, 0};
s4 = {0, 0, l1 + l2, \[Minus]1, 0, 0};
s5 = {l1 + l2, 0, 0, 0, 0, 1};
s6 = {0, 0, 0, 0, 1, 0};
q1 = {0, 0, 0};
q2 = q1;
q3 = q2;
q4 = {0, l1 , 0};
q5 = {0, l1 + l2, 0};
q6 = q5;
e1 = TwistExp[s1, q1];
e2 = TwistExp[s2, q2];
e3 = TwistExp[s3, q3];
e4 = TwistExp[s4, q4];
e5 = TwistExp[s5, q5];
e6 = TwistExp[s6, q6];
MatrixForm[gst = Simplify[e1 . e2 . e3 . e4 . e5 . e6 . g0]];
g1 = gst;
g2 = e2 . e3 . e4 . e5 . e6 . g0;
g3 = e3 . e4 . e5 . e6 . g0;
g4 = e4 . e5 . e6 . g0;
g5 = e5 . e6 . g0;
g6 = e6 . g0;
adj1 = Inverse[RigidAdjoint[g1]];
adj2 = Inverse[RigidAdjoint[g2]];
adj3 = Inverse[RigidAdjoint[g3]];
adj4 = Inverse[RigidAdjoint[g4]];
adj5 = Inverse[RigidAdjoint[g5]];
adj6 = Inverse[RigidAdjoint[g6]];
s1' = adj1 . s1;
s2' = adj1 . s2;
s3' = adj1 . s3;
 {
 {s4' = adj1 . s4;},
 {s5' = adj1 . s5;
  s6' = adj1 . s6;
  jacob = -Transpose[{s1', s2', s3', s4', s5', s6}];
  force = {fx, fy, fz, tx, ty, tz};
  MatrixForm[Torque = jacob . force]}
}


Inverse Elbow Manipulator
gst0 = {{1, 0, 0, 0}, {0, 1, 0, l1 + l2 + l3}, {0, 0, 1, 0}, {0, 0, 0,
     1}};
s1 = {0, 0, 0, 0, 0, 1};
s2 = {0, 0, 0, -1, 0, 0};
s3 = {0, 1, 0, 0, 0, 0};
s4 = {0, 0, l1 + l2, \[Minus]1, 0, 0};
s5 = {l1 + l2, 0, 0, 0, 0, 1};
s6 = {0, 0, 0, 0, 1, 0};
q1 = {0, 0, 0};
q2 = q1;
q3 = {0, l1, 0};
q4 = {0, l1 + l2, 0};
q5 = q4;
q6 = q5;
e1 = TwistExp[s1, q1];
e2 = TwistExp[s2, q2];
e3 = TwistExp[s3, q3];
```

```
e4 = TwistExp[s4, q4];
e5 = TwistExp[s5, q5];
e6 = TwistExp[s6, q6];
MatrixForm[gst = Simplify[e1 . e2 . e3 . e4 . e5 . e6 . gst0]];
g1 = gst;
g2 = e2 . e3 . e4 . e5 . e6 . g0;
g3 = e3 . e4 . e5 . e6 . g0;
g4 = e4 . e5 . e6 . g0;
g5 = e5 . e6 . g0;
g6 = e6 . g0;
adj1 = Inverse[RigidAdjoint[g1]];
adj2 = Inverse[RigidAdjoint[g2]];
adj3 = Inverse[RigidAdjoint[g3]];
adj4 = Inverse[RigidAdjoint[g4]];
adj5 = Inverse[RigidAdjoint[g5]];
adj6 = Inverse[RigidAdjoint[g6]];
s1' = adj1 . s1;
s2' = adj1 . s2;
s3' = adj1 . s3;
 {
 {s4' = adj1 . s4;},
 {s5' = adj1 . s5;
  s6' = adj1 . s6;
  jacob = -Transpose[{s1', s2', s3', s4', s5', s6}];
  force = {fx, fy, fz, tx, ty, tz};
  MatrixForm[Torque = jacob . force]}
}

Stanford Manipulator
g0 = {{1, 0, 0, 0}, {0, 1, 0, l1 + l2 + l3}, {0, 0, 1, 0}, {0, 0, 0,
    1}};
s1 = {0, 0, 0, 0, 0, 1};
s2 = {0, 0, 0, \[Minus]1, 0, 0};
s3 = {0, 1, 0, 0, 0, 0};
s4 = {0, 0, l1 + l2, \[Minus]1, 0, 0};
s5 = {l1 + l2, 0, 0, 0, 0, 1};
s6 = {0, 0, 0, 0, 1, 0};
q1 = {0, 0, 0};
q2 = q1;
q3 = {0, l1, 0};
q4 = {0, l1 + l2, 0};
q5 = q4;
q6 = q5;
e1 = TwistExp[s1, q1];
e2 = TwistExp[s2, q2];
e3 = TwistExp[s3, q3];
e4 = TwistExp[s4, q4];
e5 = TwistExp[s5, q5];
e6 = TwistExp[s6, q6];
MatrixForm[gst = Simplify[e1 . e2 . e3 . e4 . e5 . e6 . g0]];
g1 = gst;
g2 = e2 . e3 . e4 . e5 . e6 . g0;
g3 = e3 . e4 . e5 . e6 . g0;
g4 = e4 . e5 . e6 . g0;
g5 = e5 . e6 . g0;
g6 = e6 . g0;
adj1 = Inverse[RigidAdjoint[g1]];
adj2 = Inverse[RigidAdjoint[g2]];
adj3 = Inverse[RigidAdjoint[g3]];
adj4 = Inverse[RigidAdjoint[g4]];
```

```
adj5 = Inverse[RigidAdjoint[g5]];
adj6 = Inverse[RigidAdjoint[g6]];
s1' = adj1 . s1;
s2' = adj1 . s2;
s3' = adj1 . s3;
 {
 {s4' = adj1 . s4;},
 {s5' = adj1 . s5;
  s6' = adj1 . s6;
  jacob = -Transpose[{s1', s2', s3', s4', s5', s6}];
  force = {fx, fy, fz, tx, ty, tz};
  MatrixForm[Torque = jacob . force]}
}

Rhino Robot
g0 = {{1, 0, 0, 0}, {0, 1, 0, l1 + l2 + l3}, {0, 0, 1, 0}, {0, 0, 0,
    1}};
s1 = {0, 1, 0, 0, 0, 0};
s2 = {0, 0, 0, 0, 0, 1};
s3 = {0, 0, -1, 0, 0, 0};
s4 = {0, 0, l1, \[Minus]1, 0, 0};
s5 = {0, 0, l1 + l2, -1, 0, 0};
s6 = {0, 0, 0, 0, 1, 0};
q1 = {0, 0, 0};
q2 = q1;
q3 = q2;
q4 = {0, l1, 0};
q5 = {0, l1 + l2, 0};
q6 = q5;
e1 = TwistExp[s1, q1];
e2 = TwistExp[s2, q2];
e3 = TwistExp[s3, q3];
e4 = TwistExp[s4, q4];
e5 = TwistExp[s5, q5];
e6 = TwistExp[s6, q6];
MatrixForm[gst = Simplify[e1 . e2 . e3 . e4 . e5 . e6 . g0]];
g1 = gst;
g2 = e2 . e3 . e4 . e5 . e6 . g0;
g3 = e3 . e4 . e5 . e6 . g0;
g4 = e4 . e5 . e6 . g0;
g5 = e5 . e6 . g0;
g6 = e6 . g0;
adj1 = Inverse[RigidAdjoint[g1]];
adj2 = Inverse[RigidAdjoint[g2]];
adj3 = Inverse[RigidAdjoint[g3]];
adj4 = Inverse[RigidAdjoint[g4]];
adj5 = Inverse[RigidAdjoint[g5]];
adj6 = Inverse[RigidAdjoint[g6]];
s1' = adj1 . s1;
s2' = adj1 . s2;
s3' = adj1 . s3;
 {
 {s4' = adj1 . s4;},
 {s5' = adj1 . s5;
  s6' = adj1 . s6;
  jacob = -Transpose[{s1', s2', s3', s4', s5', s6}];
  force = {fx, fy, fz, tx, ty, tz};
  MatrixForm[Torque = jacob . force]}
}
```

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

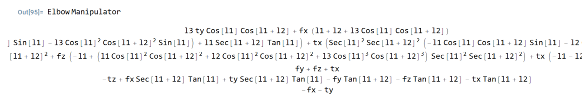


Figure 3: Result for Elbow Manipulator

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 $(\alpha, \beta, \gamma)$ 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 $q_{1,\ldots}, q_n$, 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:

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

***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.

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

   where $\tau_k$ is the input generalized force.

   **Proof 1:**

   Euler-Lagrange equation with external force $\tau_k$ 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 $D(q) = \begin{bmatrix} d_{11}(q) & d_{12}(q) \\ d_{12}(q) & d_{22} \end{bmatrix}$. 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.

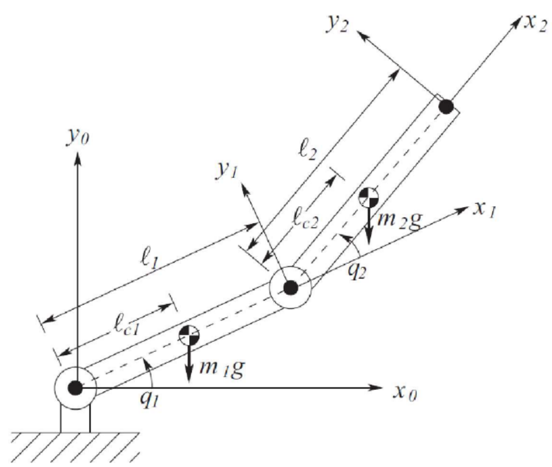


Figure 4: Two-link revolute joint arm. The rotational joint motion introduces dynamic coupling between the joints. (Figure 6.9 in SHV)

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

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