Robotics II
Day 8: Jacobian

Chua Tzong Lin
B7TB1703
chuatl@mems.mech.tohoku.ac.jp

1. Calculate the "Jacobian" for the manipulator in the given file "Velocity_Jacobian_3Links_3D.py" and complete the python script.

   The forward kinematics of the system is given as:

$$X_0 = 0$$
$$Y_0 = 0$$
$$Z_0 = 0$$

$$X_1 = 0$$
$$Y_1 = 0$$
$$Z_1 = L_0$$

$$X_2 = L_1 \cos(\theta_2)\cos(\theta_1)$$
$$Y_2 = L_1 \cos(\theta_2)\sin(\theta_1)$$
$$Z_2 = L_1 \sin(\theta_2) + Z_1$$

$$X_3 = X_1 + L_2 \cos(\theta_2 + \theta_3)\cos(\theta_1)$$
$$Y_3 = Y_1 + L_2 \cos(\theta_2 + \theta_3)\sin(\theta_1)$$
$$Z_3 = Z_1 + L_2 \sin(\theta_2 + \theta_3)$$

   By partial differentiating the forward kinematics equation with respect to θ_1, θ_2, and θ_3, the Jacobian of the system is constructed using the following equation.

$$\mathbf{J} = \begin{bmatrix} \frac{\partial \mathbf{f}}{\partial x_1} & \cdots & \frac{\partial \mathbf{f}}{\partial x_n} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

   The elements of the Jacobian matrix are obtained as the following.

$$J_{11} = [-L_1 \cos(\theta_2) - L_2 \cos(\theta_2 + \theta_3)]\sin(\theta_1)$$
$$J_{12} = [-L_1 \sin(\theta_2) - L_2 \sin(\theta_2 + \theta_3)]\cos(\theta_1)$$
$$J_{13} = -L_2 \sin(\theta_2 + \theta_3)\cos(\theta_1)$$

$$J_{21} = [L_1 \cos(\theta_2) + L_2 \cos(\theta_2 + \theta_3)]\cos(\theta_1)$$
$$J_{22} = [-L_1 \sin(\theta_2) - L_2 \sin(\theta_2 + \theta_3)]\sin(\theta_1)$$
$$J_{23} = -L_2 \sin(\theta_2 + \theta_3)\sin(\theta_1)$$

$$J_{31} = 0$$
$$J_{32} = L_1 \cos(\theta_2) + L_2 \cos(\theta_2 + \theta_3)$$
$$J_{33} = L_2 \cos(\theta_2 + \theta_3)$$

The implemented python script is included under the filename Velocity_Jacobian_3Links_3D.py.

The result is shown in the following figure 1.1.

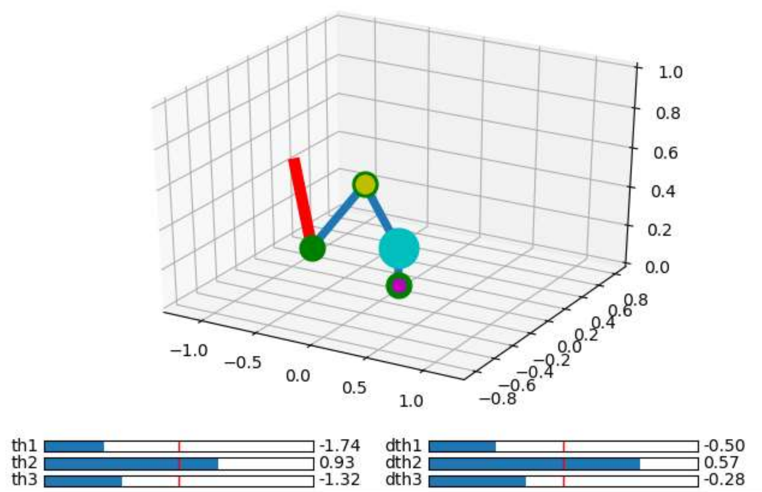

Figure 1.1 Velocity Jacobian for 3 Link Mechanism in 3D

2. Trajectory tracking for a redundant manipulator. (The target trajectory in the given script Trajectory_Tracking_Jacobian_5Links.py is required to be changed in to a nonlinear path.)

   The given 5 link manipulator is manipulated in a 2D plane. Therefore the given foreward kinematics is given by the following equations.

   The forward kinematics of a 5 Link Mechanism in 2D is given as:

$$X_1 = L_1 \cos(\theta_1)$$
$$Y_1 = L_1 \sin(\theta_1)$$

$$X_2 = L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2)$$
$$Y_2 = L_1 \sin(\theta_1) + L_2 \sin(\theta_1 + \theta_2)$$

$$X_3 = L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2) + L_3 \cos(\theta_1 + \theta_2 + \theta_3)$$
$$Y_3 = L_1 \sin(\theta_1) + L_2 \sin(\theta_1 + \theta_2) + L_3 \sin(\theta_1 + \theta_2 + \theta_3)$$

$$X_4 = L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2) + L_3 \cos(\theta_1 + \theta_2 + \theta_3) + L_4 \cos(\theta_1 + \theta_2 + \theta_3 + \theta_4)$$
$$Y_4 = L_1 \sin(\theta_1) + L_2 \sin(\theta_1 + \theta_2) + L_3 \sin(\theta_1 + \theta_2 + \theta_3) + L_4 \sin(\theta_1 + \theta_2 + \theta_3 + \theta_4)$$

$$X_5 = L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2) + L_3 \cos(\theta_1 + \theta_2 + \theta_3) + L_4 \cos(\theta_1 + \theta_2 + \theta_3 + \theta_4) + L_5 \cos(\theta_1 + \theta_2 + \theta_3 + \theta_4 + \theta_5)$$
$$Y_5 = L_1 \sin(\theta_1) + L_2 \sin(\theta_1 + \theta_2) + L_3 \sin(\theta_1 + \theta_2 + \theta_3) + L_4 \sin(\theta_1 + \theta_2 + \theta_3 + \theta_4) + L_5 \sin(\theta_1 + \theta_2 + \theta_3 + \theta_4 + \theta_5)$$

In the case of a redundant manipulator, the dimension of the position vector of the end point of the manipulator in the 2D case is 2, however in the case of a 5 link manipulator moving in a 2D plane, the degree of freedom of the system is 5. Therefore, the Jacobian of the system is mapping a 5 dimensional vector to a 2nd dimensional vector, hence the Jacobian matrix is a 2 X 5 matrix with a rank of 2.

The elements Jacobian of the system is given as:

$$\begin{aligned} X_5 = L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2) + L_3 \cos(\theta_1 + \theta_2 + \theta_3) + L_4 \cos(\theta_1 + \theta_2 + \theta_3 + \theta_4) \\ + L_5 \cos(\theta_1 + \theta_2 + \theta_3 + \theta_4 + \theta_5) \end{aligned}$$

$$\begin{aligned} J_{11} &= -L_1 \sin(\theta_1) - L_2 \sin(\theta_1 + \theta_2) - L_3 \sin(\theta_1 + \theta_2 + \theta_3) - L_4 \sin(\theta_1 + \theta_2 + \theta_3 + \theta_4) \\ &\quad - L_5 \sin(\theta_1 + \theta_2 + \theta_3 + \theta_4 + \theta_5) \\ J_{12} &= -L_2 \sin(\theta_1 + \theta_2) - L_3 \sin(\theta_1 + \theta_2 + \theta_3) - L_4 \sin(\theta_1 + \theta_2 + \theta_3 + \theta_4) \\ &\quad - L_5 \sin(\theta_1 + \theta_2 + \theta_3 + \theta_4 + \theta_5) \\ J_{13} &= -L_3 \sin(\theta_1 + \theta_2 + \theta_3) - L_4 \sin(\theta_1 + \theta_2 + \theta_3 + \theta_4) \\ &\quad - L_5 \sin(\theta_1 + \theta_2 + \theta_3 + \theta_4 + \theta_5) \\ J_{14} &= - - L_4 \sin(\theta_1 + \theta_2 + \theta_3 + \theta_4) - L_5 \sin(\theta_1 + \theta_2 + \theta_3 + \theta_4 + \theta_5) \\ J_{15} &= -L_5 \sin(\theta_1 + \theta_2 + \theta_3 + \theta_4 + \theta_5) \end{aligned}$$

$$\begin{aligned} J_{21} &= L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2) + L_3 \cos(\theta_1 + \theta_2 + \theta_3) + L_4 \cos(\theta_1 + \theta_2 + \theta_3 + \theta_4) \\ &\quad + L_5 \cos(\theta_1 + \theta_2 + \theta_3 + \theta_4 + \theta_5) \\ J_{22} &= L_2 \cos(\theta_1 + \theta_2) + L_3 \cos(\theta_1 + \theta_2 + \theta_3) + L_4 \cos(\theta_1 + \theta_2 + \theta_3 + \theta_4) \\ &\quad + L_5 \cos(\theta_1 + \theta_2 + \theta_3 + \theta_4 + \theta_5) \\ J_{23} &= L_3 \cos(\theta_1 + \theta_2 + \theta_3) + L_4 \cos(\theta_1 + \theta_2 + \theta_3 + \theta_4) \\ &\quad + L_5 \cos(\theta_1 + \theta_2 + \theta_3 + \theta_4 + \theta_5) \\ J_{24} &= L_4 \cos(\theta_1 + \theta_2 + \theta_3 + \theta_4) + L_5 \cos(\theta_1 + \theta_2 + \theta_3 + \theta_4 + \theta_5) \\ J_{25} &= L_5 \cos(\theta_1 + \theta_2 + \theta_3 + \theta_4 + \theta_5) \end{aligned}$$

Since the Jacobian is not a square matrix, the inverse of the matrix does not exist, hence there exist multiple solutions to the velocity equation. In order to obtain a unique solution, the pseudo-inverse of the Jacobian matrix is calculated, which can be considered as a type of generalized inverse of a matrix.

The script of for the trajectory tracking for a redundant is included under the filename Trajectory_Tracking_Jacobian_5Links.py.

The script is modified in a manner that the program will initially prompt the user for a desired polynomial to track, after inputting the desired polynomial degree, the user will be have an option of manual inputting the polynomial coefficients or randomly generating the input.

The procedure is shown in the following figures.

1st Prompt:

```
Input Desired Trajectory Degree (Integers only):
```

2nd Prompt:

```
Randomize inputs? y/n
```

Input sample for randomize inputs

```
Input Desired Trajectory Degree (Integers only): 7
Randomize inputs? y/n
y
```

Input sample manually inputted coefficients

```
Input Desired Trajectory Degree (Integers only): 7
Randomize inputs? y/n
n
(Float or integers only)a[1]: 2
(Float or integers only)a[2]: 4
(Float or integers only)a[3]: 6
(Float or integers only)a[4]: 8
(Float or integers only)a[5]: 10
(Float or integers only)a[6]: 11
(Float or integers only)a[7]: 8
```

The result is shown in the following figure 2.1.

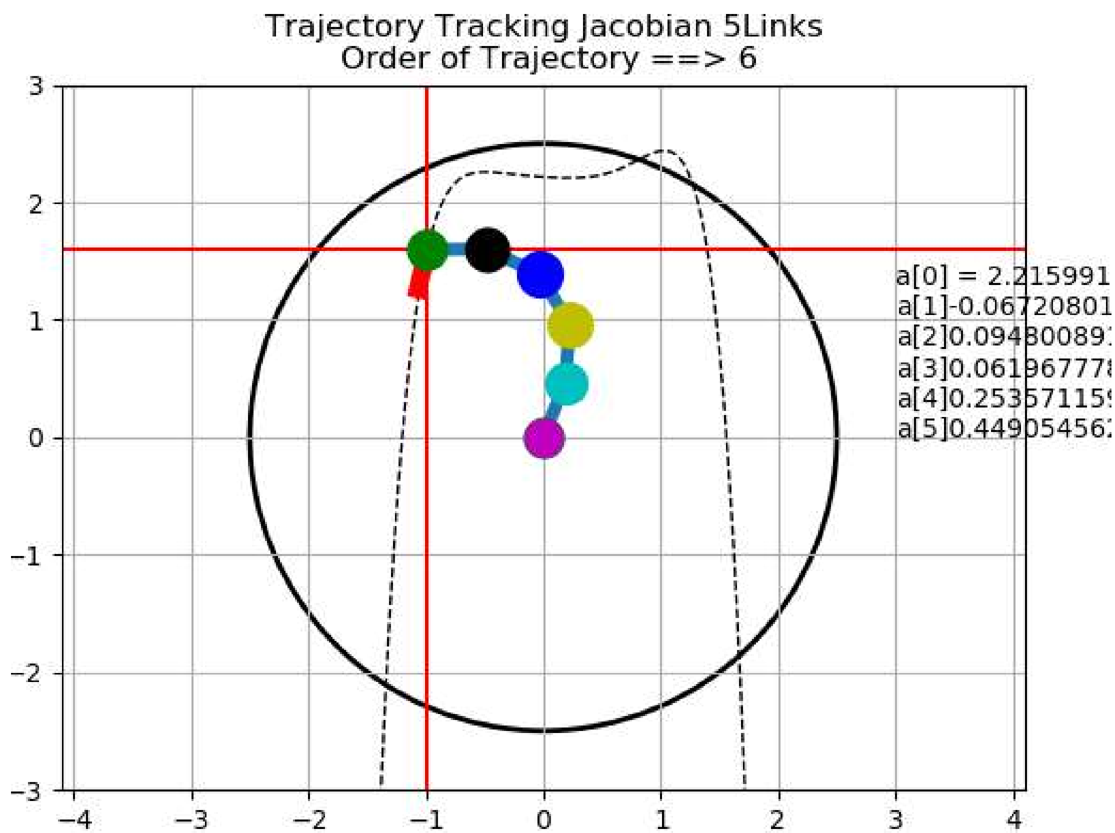

Figure 2.1 Velocity trajectory of a 5 link manipulator