Lab assignment 1

EE 183

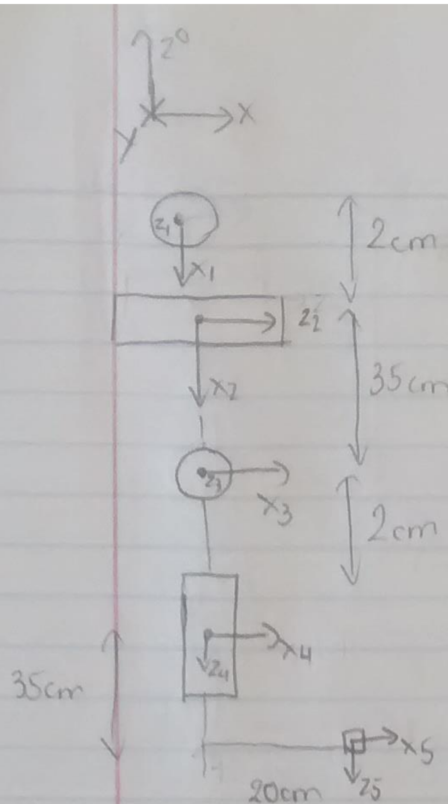
Minh Cao – 804844568

Jan 25th 2018

Introduction:

Human is among the best animal that is capable of running marathon or even ultramarathon. A combination of sweat gland and especially how leg joints is what enable us to do this. In this lab, we want to model human leg with 4 rotational kinematic linkages – 2 at the hips and 2 and the ankle and the toes as the end-effector – to approximate human motion.

The implementation of the leg will be described by the Denavit–Hartenberg parameter of the links between joints. We will feed it to the corresponding forward kinematic matrix to figure out the position of the end-effector relative to the hips at any states of the linkage. From the matrix, we will approximate numerically inverse kinematic matrix so that we can figure out the states of the linkage necessary to put the end-effector to the desire position. This simple model will give us the base to develop more complex model that will eventually capable of modeling all the dynamic of all joints during human motion and potentially study that motion from normal human motion to develop efficient bipedal robot.



		a	α	d	θ	Initial
Link	1	0	90°	0	θ_1	90°
	2	2	-90°	0	θ_2	0°
	3	35	90°	0	θ_3	90°
	4	0	90°	2	θ_4	0°
	5	20	0	35	0	

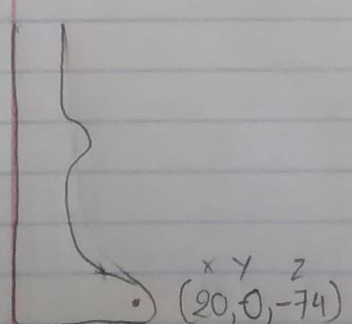
Forward kinematic matrix

$$T = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & \alpha_{i-1} \\ \sin\theta_i\cos\alpha_{i-1} & \cos\theta_i\cos\alpha_{i-1} & -\sin\alpha_{i-1} & -\sin\alpha_{i-1}\, d_i \\ \sin\theta_n\sin\alpha_{i-1} & \cos\theta_i\sin\alpha_{i-1} & \cos\alpha_{i-1} & \cos\alpha_{i-1}\, d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

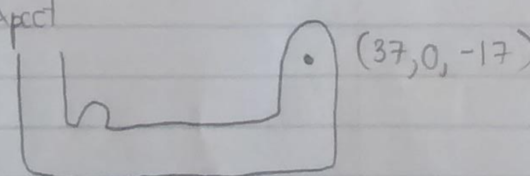
Initial $p\,[90^\circ\ 0^\circ\ 90^\circ\ 0^\circ]$

Broken bone
kick position p ?

Expect



Expect



Method:

D-H parameter:

Setting up Denavit–Hartenberg for all the linkage is crucial step before doing any forward or inverse kinematic. In this lab, I use the modified version of D-H parameter which has these rule:

- $\mathbf{a_{i-1}}$ is the distance between $\mathbf{O_{i-1}}$ and $\mathbf{O_i}$ along x_{i-1} axis.
- $\boldsymbol{\alpha_{i-1}}$ is the angle to align z_{i-1} to z_i along x_{i-1} axis.
- $\mathbf{d_i}$ is the distance between $\mathbf{O_{i-1}}$ and $\mathbf{O_i}$ along z_{i-1} axis.
- $\boldsymbol{\theta_i}$ is the angle to align x_{i-1} to x_i along z_{i-1} axis.

Forward kinematic:

$$^{n-1}T_n = \left[\begin{array}{ccc|c} \cos\theta_n & -\sin\theta_n & 0 & a_{n-1} \\ \sin\theta_n\cos\alpha_{n-1} & \cos\theta_n\cos\alpha_{n-1} & -\sin\alpha_{n-1} & -d_n\sin\alpha_{n-1} \\ \sin\theta_n\sin\alpha_{n-1} & \cos\theta_n\sin\alpha_{n-1} & \cos\alpha_{n-1} & d_n\cos\alpha_{n-1} \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

(source: https://en.wikipedia.org/wiki/Denavit%E2%80%93Hartenberg_parameters#Modified_DH_parameters)

The final matrix T_i^0 from end effector view to global view will be the product of all transform matrix along the way. The final state in global view will be:

$$p^0 = T_i^0 p^{end-effector}$$

With p is the state at particular view, which is a 4x1 vector contain x, y, z position and 1 at the end.

Inverse kinematic:

While forward kinematic let us know the position in global view given position at end-effector view and the states of all the linkage. I would be more useful if we can find out the states of all linkage given position at end-effector and global view. To do this, inverse kinematic is used with these simple principles:

$$x = f(q)$$

$$\frac{dx}{dt} = \frac{df(q)}{dt} = J \times \frac{dq}{dt}$$

Where J is the Jacobian matrix with give the rate of change for every component of x cause by every component of q.

We can calculate numerically by adding small step at every component of current q and use forward kinematics to calculate the corresponding x result from that change. We can then construct the Jacobian matrix from the result. We can then take a small step from current position to desire to position and convert them to small step in states using the inverse Jacobian matrix.

The inverse of the equation above is:

$$\frac{dq}{dt} = \frac{df^{-1}(x)}{dt} = J^{-1} \times \frac{dx}{dt}$$

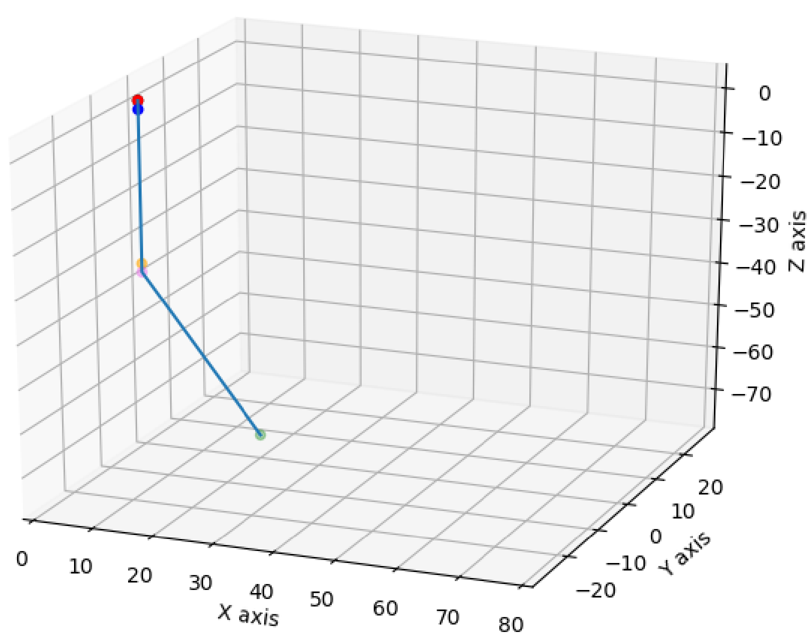
We can then add that dq to our current state and repeat the step until we reach the position that we desired. We will get the final state q for that position.

Note: Since J is most likely not a square matrix, normal inverse matrix will not possible. Pseudo-matrix inversion will be used here with the formula:

$$J^{+} = J^{T}(J\,J^{T})^{-1}$$

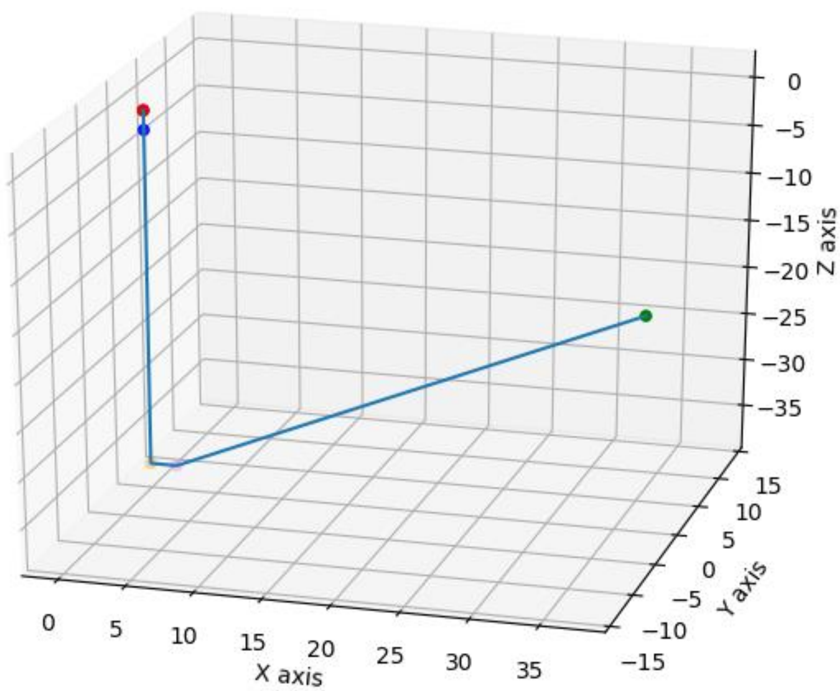
Result:

We expect output for initial state at end-effector p = [-90 0 90 0] to be similar to a standing leg.



Output position in global view is [20, 0, -74] as we expected

Now we want to move it to position [37, 0, -17] which is resemble to the broken leg position with the feet upright. Plugging in the current state and the desire position we get:



The states of the joints angle to achieve this position is [-90 -0.28 180 -0.21]. Some error deviate from our expected [-90 0 -180 0] comes from the process of inverse matrix and the acceptable norm difference we set at 0.0001

Also from the result above, the third parameter change from 90 to 180 as we expected since the knee portion swing up to the desire position. This is a curve trajectory, not straight-line motion in operational space. To create a straight line trajectory, one would need to implement the same process for many points in that straight-line path to achieve it by moving many joints correspondingly, not just a single one like this example.

This lab takes me around 10-15 hours to complete. It was hard but when the inverse kinematic code works, it feels like magic.

Code python: https://github.com/mqc25/EE183/blob/master/Lab1.py

Reference:
https://en.wikipedia.org/wiki/Denavit%E2%80%93Hartenberg_parameters#Modified_DH_parameters