로봇동역학 및 제어

Robot Dynamics & Control

최 종 보 고 서

기계설계로봇공학과 19510053 김지용

2019. 06. 23.

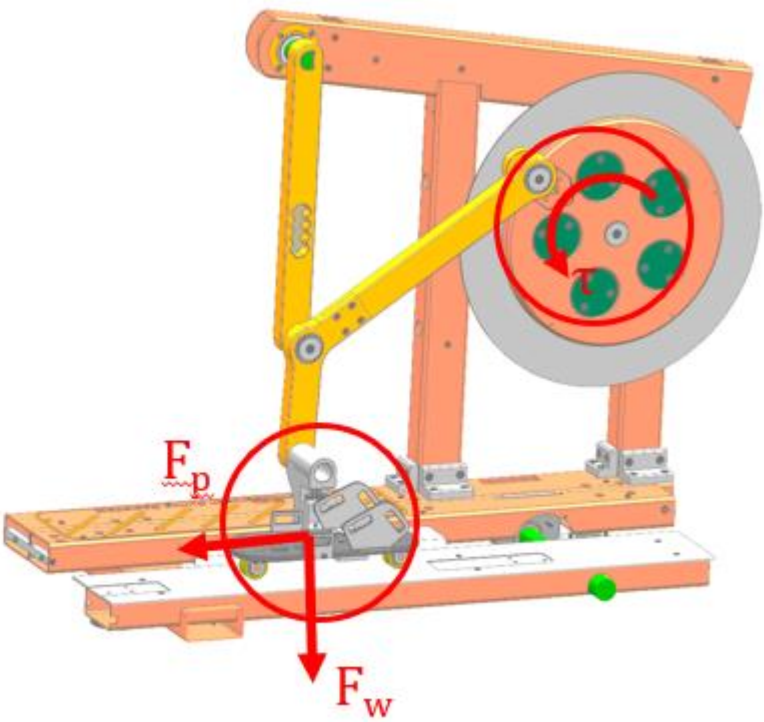
순　　　서

1. 연구 배경 및 필요성

인구 고령화로 인한 **노년층 장애 환자**가 증가하고 사고로 인한 **마비 환자**들이 지속적으로 발생하고 있다. 이는 로봇의 발전과 함께 다양한 종류의 재활로봇이 연구 및 개발되고 있는 상황이다. 특히, 보행과 관련된 **보행재활로봇**에 대한 연구가 활발하게 진행되고 있는데, 로봇의 도움으로 하지 마비 환자들에게 필요한 운동을 수행할 수 있게 도와주는 로봇들이 개발되고 있다.

하지 마비 환자들은 **장기간의 지속적인 운동**이 필요한데, 스스로 재활할 수 있는 능력이 제한되는 하지 마비 환자들에게는 보행재활로봇이 효과적인 치료요소가 되고 있다. 이러한 보행재활로봇의 재활 효과를 높이기 위해서는 로봇이 **사람의 보행 특성을 반영한 운동**을 해야 하며, 이를 위한 기구 개발 및 알고리즘 개발이 필수적이다.

2. 작업 목표

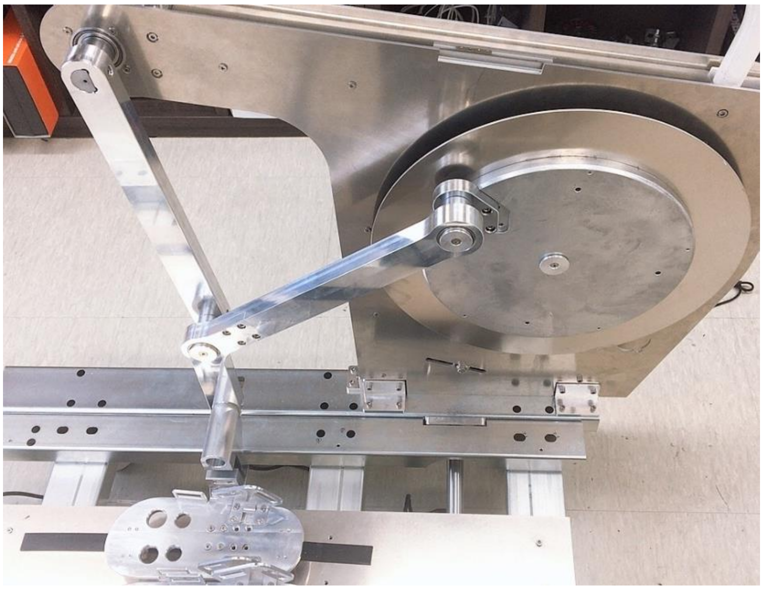


<그림 1> 작업 목표

작업 목표는 환자가 로봇에 탑승했을 때 **환자에 의해 로봇의 발판에 걸리는 힘**(사람의 하중과 운동하는 힘)과 **크랭크 축에 걸리는 토크 사이의 관계**를 구하는 것이다. 이 관계를 통해 보행 재활 운동 시 발판의 위치와 환자의 힘(의도)에 따라 필요한 모터의 토크 크기를 구하여 효과적인 운동을 만든다.

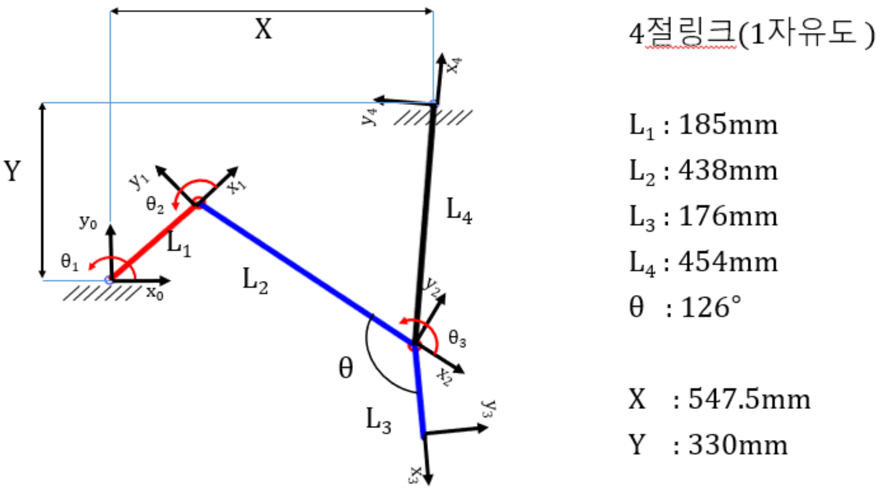
3. 로봇사양

<그림 2>는 실제 로봇의 4절링크 부분 사진이다.



<그림 2> 실제 로봇 4절 링크

이를 단순화하여 도식하면 <그림 3>과 같다.

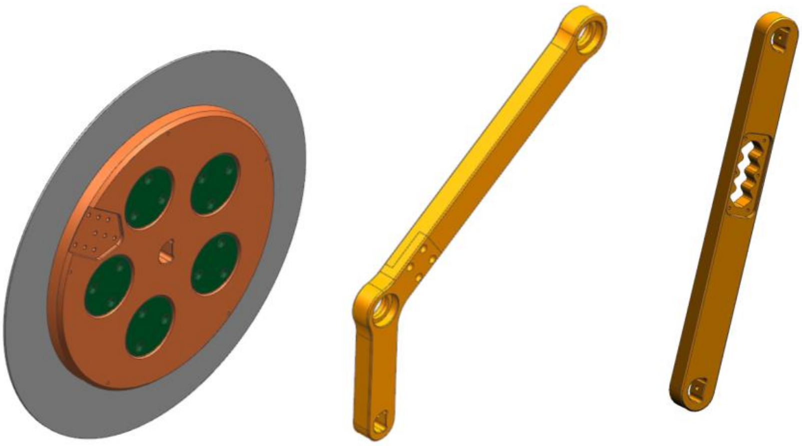


<그림 3> 4절링크 도식화

각 링크의 길이는 실제 링크의 길이를 기준으로 하였으며, 각각의 링크 Joint 에 Joint number 를 부여하였으며, 원점을 Joint 1 으로 잡고 **Cartesian 좌표계**를 적용하였다.

이 로봇의 특이점으로 **링크 4 에 의한 Constraint** 이 있다는 것이다. (x_4, y_4)의 위치는 고정되어 있으며, 원점으로부터의 거리는 (547.5, 330) 이다.

각 링크의 형태는 <그림 4>와 같으며, 3D 모델링 상에서 얻어온 형상이다. 3D 모델링을 통해서 각 링크의 데이터를 얻을 수 있으며, 이 프로젝트에서는 단순히 무게만 사용하였다.



<그림 4> 각 링크의 3D 모델

각각의 무게는 7.7kg, 2.5kg, 2.4kg 이다.

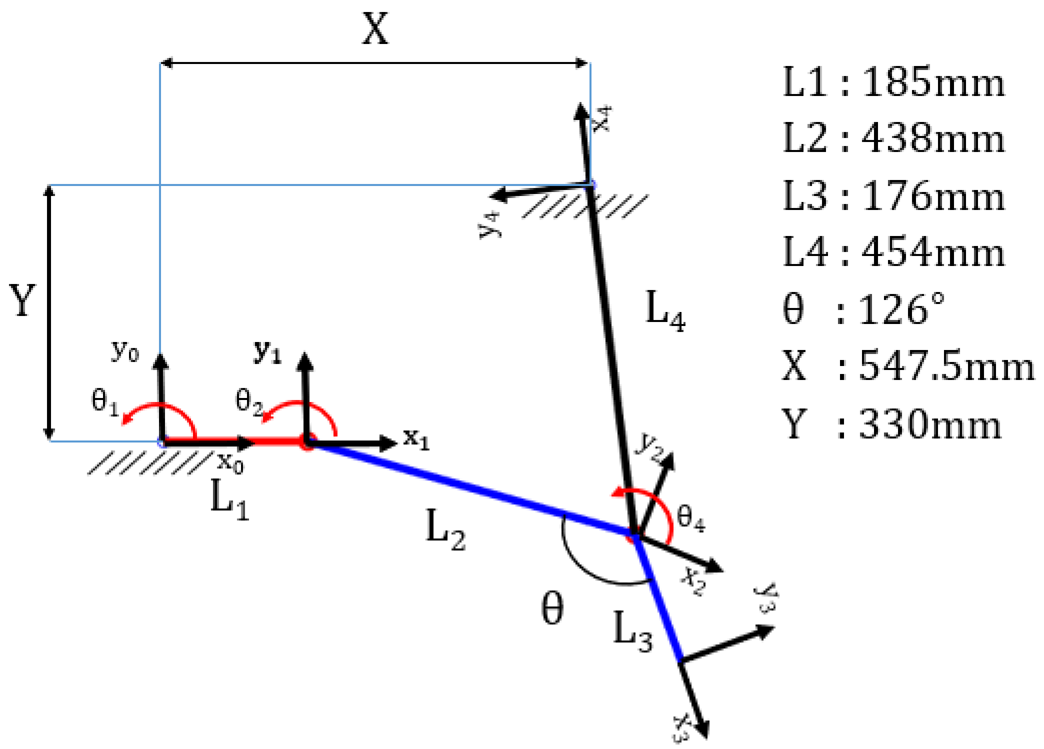
나중에 심화해서 연구할 때는 링크의 무게 뿐만 아니라 질량중심, 질량 관성 모멘트 등과 같은 데이터를 사용하면 조금 더 정확한 결과를 얻을 수 있을 것이다.

4. 순기구학(F.K.)

F.K.를 풀기 위해서 먼저 로봇의 초기 위치를 정하고, D-H Convention을 만족하도록 좌표를 선정하였다. 그리고 D-H Convention을 이용해 각각의 Homogeneous matrix를 구하였다.

<그림 5>는 초기 위치인 θ_1을 0으로 두었을 때 모습이다. 그런데 초기위치일 때 θ_2가 Constraint에 의해 -15.9233° 기울어져 있으며, θ_4 또한 113.3539° 기울어져 있음을 알

수 있다. 그리고 링크 2번과 3번은 원래 하나의 링크이지만, 계산의 편의성을 위해 두 부분으로 나누었고, 이때 Joint variable은 -54°로 고정시켰다.



<그림 5> D-H Convention을 만족하는 로봇의 초기 위치

D-H convention을 풀기 위해 Parameter들을 <표 1>과 같이 구하였다.

	a	α	d	θ
1	L_1	0	0	θ_1
2	L_2	0	0	θ_2-15.9233
3	L_3	0	0	θ-180
4	L_4	0	0	θ_4+113.3539

<표 1> D-H Convention Parameter Table

D-H Convention parameter을 이용하여 Homogeneous matrix를 다음과 같이 구하였다. Planar Robot으로 **z축 이동과 회전은 고려되지 않음**을 알 수 있다.

$$A_1 = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & L_1 \cdot \cos\theta_1 \\ \sin\theta_1 & \cos\theta_1 & 0 & L_1 \cdot \sin\theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

$$A_2 = \begin{bmatrix} \cos\theta_2^* & -\sin\theta_2^* & 0 & L_2 \cdot \cos\theta_2^* \\ \sin\theta_2^* & \cos\theta_2^* & 0 & L_2 \cdot \sin\theta_2^* \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

$$A_3 = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & L_3 \cdot \cos\theta_3 \\ \sin\theta_3 & \cos\theta_3 & 0 & L_3 \cdot \sin\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

$$A_4 = \begin{bmatrix} \cos\theta_4^* & -\sin\theta_4^* & 0 & L_2 \cdot \cos\theta_4^* \\ \sin\theta_4^* & \cos\theta_4^* & 0 & L_2 \cdot \sin\theta_4^* \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

위의 4 개의 Homogeneous matrix 들을 이용하여 다음과 같이 T-matrix 들을 구하였다.

(6)

$$T_2^0 = A_1 \cdot A_2 = \begin{bmatrix} \cos(\theta_1 + \theta_2^*) & -\sin(\theta_1 + \theta_2^*) & 0 & L_1 \cdot \cos\theta_1 + L_2 \cdot \cos(\theta_1 + \theta_2^*) \\ \sin(\theta_1 + \theta_2^*) & \cos(\theta_1 + \theta_2^*) & 0 & L_1 \cdot \sin\theta_1 + L_2 \cdot \sin(\theta_1 + \theta_2^*) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(7)

$$T_3^0 = A_1 \cdot A_2 \cdot A_3$$
$$= \begin{bmatrix} \cos(\theta_1 + \theta_2^* + \theta_3) & -\sin(\theta_1 + \theta_2^* + \theta_3) & 0 & L_1 \cdot \cos\theta_1 + L_2 \cdot \cos(\theta_1 + \theta_2^*) + L_3 \cdot cos(\theta_1 + \theta_2^* + \theta_3) \\ \sin(\theta_1 + \theta_2^* + \theta_3) & \cos(\theta_1 + \theta_2^* + \theta_3) & 0 & L_1 \cdot \sin\theta_1 + L_2 \cdot \sin(\theta_1 + \theta_2^*) + L_3 \cdot sin(\theta_1 + \theta_2^* + \theta_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(8)

$$T_4^0 = A_1 \cdot A_2 \cdot A_4$$
$$= \begin{bmatrix} \cos(\theta_1 + \theta_2^* + \theta_4^*) & -\sin(\theta_1 + \theta_2^* + \theta_4^*) & 0 & L_1 \cdot \cos\theta_1 + L_2 \cdot \cos(\theta_1 + \theta_2^*) + L_4 \cdot cos(\theta_1 + \theta_2^* + \theta_4^*) \\ \sin(\theta_1 + \theta_2^* + \theta_4^*) & \cos(\theta_1 + \theta_2^* + \theta_4^*) & 0 & L_1 \cdot \sin\theta_1 + L_2 \cdot \sin(\theta_1 + \theta_2^*) + L_4 \cdot sin(\theta_1 + \theta_2^* + \theta_4^*) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

이 T-matrix 를 구함으로 나중에 Rotation matrix 나 x, y 좌표를 쉽게 확인할 수 있다.

다음은 각각의 x, y 좌표를 나타낸 것이며, 이 때 x_4, y_4 의 좌표는 (x, y)로 고정된 점이다.

$$x_1 = L_1 \cdot cos\,\theta_1$$
$$y_1 = L_1 \cdot \sin\theta_1 \tag{9}$$

$$x_2 = L_1 \cdot \cos\theta_1 + L_2 \cdot \cos(\theta_1 + \theta_2^*)$$
$$y_2 = L_1 \cdot \sin\theta_1 + L_2 \cdot \sin(\theta_1 + \theta_2^*) \tag{10}$$

$$x_3 = L_1 \cdot \cos\theta_1 + L_2 \cdot \cos(\theta_1 + \theta_2^*) + L_3 \cdot cos(\theta_1 + \theta_2^* + \theta_3)$$
$$y_3 = L_1 \cdot \sin\theta_1 + L_2 \cdot \sin(\theta_1 + \theta_2^*) + L_3 \cdot sin(\theta_1 + \theta_2^* + \theta_3) \tag{11}$$

여기서 θ_2 와 θ_4 는 초기 위치에서 오프셋이 있기 때문에 다음과 같이 θ_2^*와 θ_4^*로 치환하여 계산을 진행하였다.

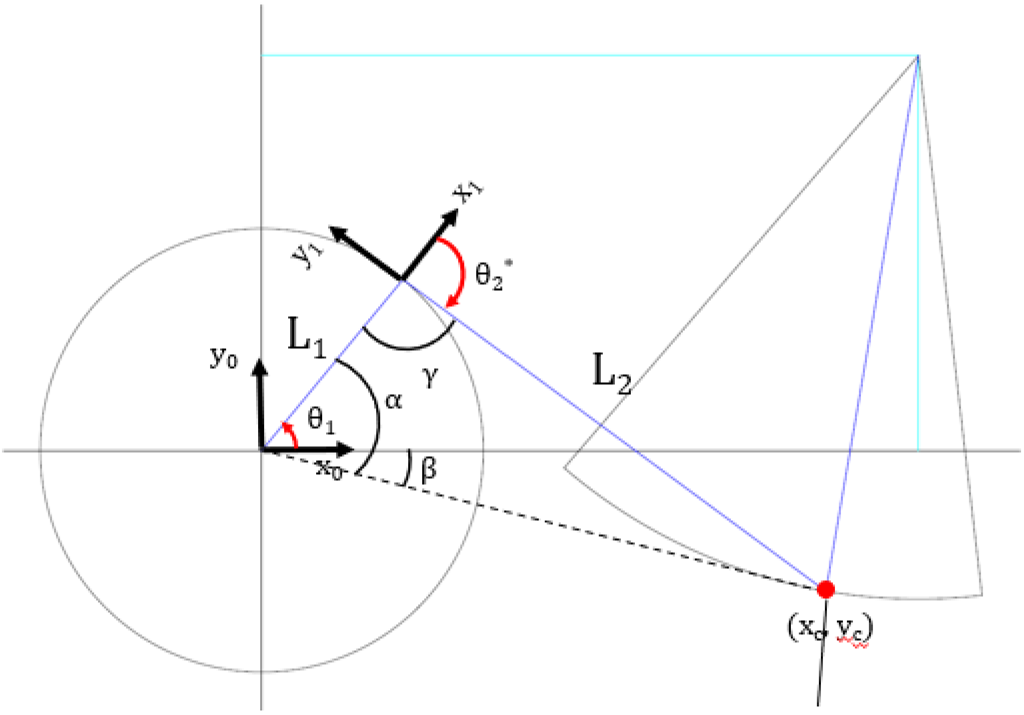
$$\theta_2^* : \theta_2 - 15.9233\,°$$
$$\theta_3 : -54° \tag{12}$$
$$\theta_4^* : \theta_4 + 113.3539\,°$$

5. 역기구학(I.K.)

역기구학을 풀기위해 전략을 세웠다. 링크 2와 링크 3 간의 Joint variable이 없기 때문에 Homogenous Vector는 일정하다. 그래서 **O_2를 End Effector의 Wrist**라고 생각하고 풀게 되었다. 따라서 O_0와 $O_2(X_c, Y_c)$ 사이의 I.K.를 구하고, 이를 구하게 되면 자연스럽게 O_2와 End Effector의 Homogenous Vector를 통해 End Effector의 위치를 구할 수 있을 것이다.

따라서 I.K.에서는 링크 4에 의한 Constraint을 고려하여 θ_1과 θ_2사이의 관계, O_0와 O_2 사이의 관계를 구하고, 그 이후 O_3 를 구하게 되었다.

<그림 6>은 I.K.를 풀 때 로봇에 필요한 Parameter들을 나타낸 것이다.



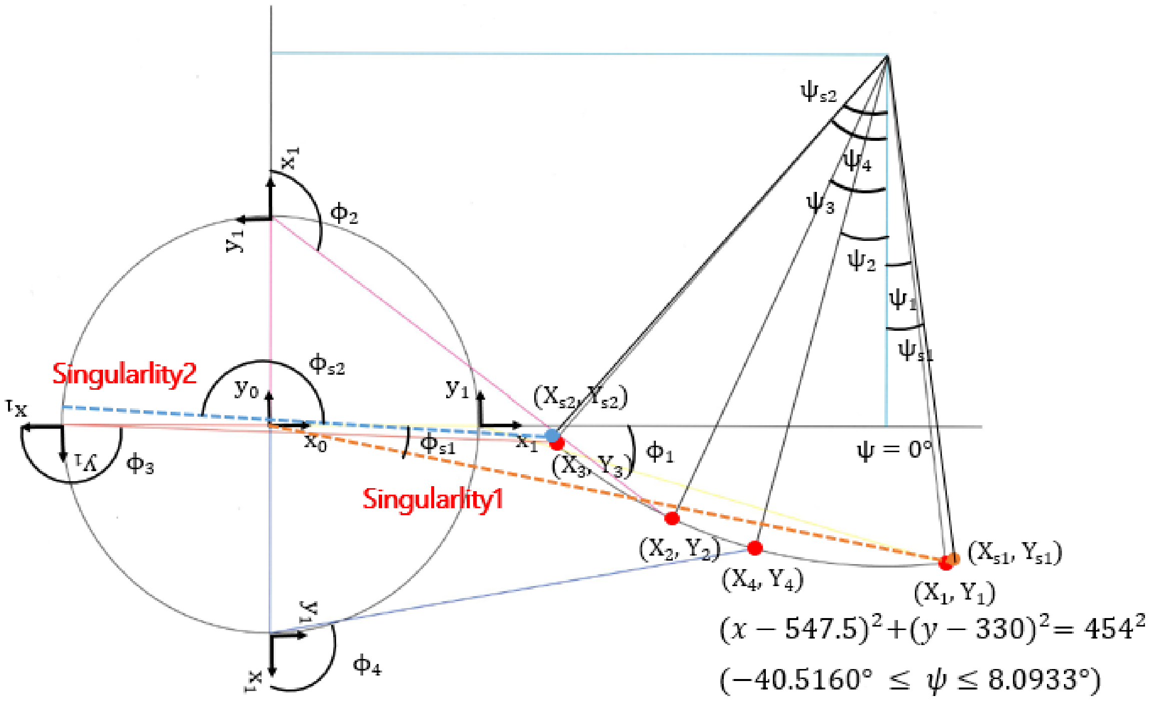
<그림 6> D-H Convention을 만족하는 로봇의 초기 위치

I.K.를 풀기에 앞서 Constraint에 의한 조건들을 먼저 살펴보았다. 이 Constraint에 의한 조건들을 통해 로봇의 전체적인 움직임을 파악할 수 있었고, 이 조건들을 이용하여 추후에 F.K.와 I.K.를 검증하는데 활용할 수 있었다.

<그림 7>은 Constraint에 의한 조건에서 각각의 값들을 구하기 위해 도식한 그림이다. θ_1이 각각 0°, 90°, 180°, -90°일 때 나머지 링크의 각도와, (X_c, Y_c)의 위치를 구하였으며, (X_c, Y_c)가 원점과 가장 가까울 때와 가장 멀 때는 Singularlity에 걸렸을 때임을 알 수 있었다. 로봇의 링크 1이 한 바퀴 돌 때 **Singularity**는 총 두 번 걸리게 되며, 이 때의 θ_1의 각도와 (X_c, Y_c)의 좌표를 구하였다.

또한 링크 4의 길이가 일정하기 때문에 (X_c, Y_c)는 **(547.5, 330)을 중심으로 하는 원의 방**

정식에 구속되어 있다.

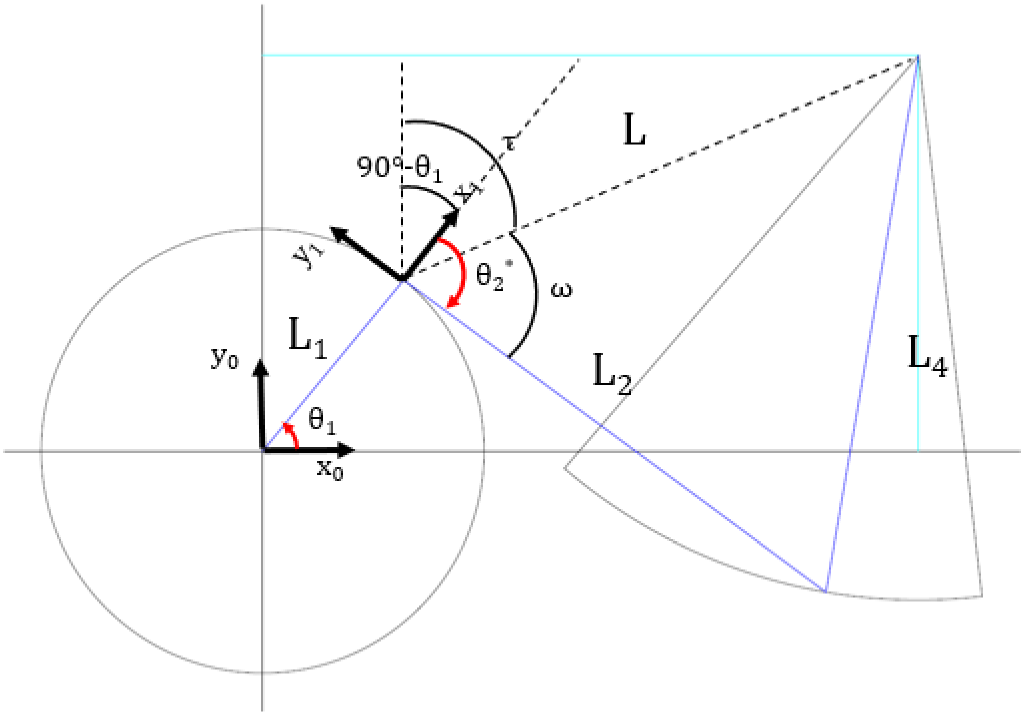


<그림 7> Constraint에 의한 조

(13)은 Constraint을 풀었을 때 나오는 각각의 값들이다. 이 값들을 구하기 위해서는 Geometric하게 풀어야 하며, 그 방법은 선을 추가로 그어서 구할 수 있으나, 여기서 그 과정은 생략한다.

$$
\begin{array}{ll}
\phi_1 : 15.9233° & X_1 : 606.1939 \\
\phi_2 : 127.0106° & Y_1 : -120.1654 \\
\phi_3 : 177.9927° & X_2 : 349.7536 \\
\phi_4 : 100.0099° & Y_2 : -78.6597 \\
\phi_{s1} : 11.0531° & X_3 : 252.7312 \\
\phi_{s2} : 176.5663° & Y_3 : -15.3418 \\
\psi_1 : 7.4302° & X_4 : 431.3326 \\
\psi_2 : 14.8226° & Y_4 : -108.8676 \\
\psi_3 : 26.7107° & X_{s1} : 611.4433 \\
\psi_4 : 40.4905° & Y_{s1} : -119.4407 \\
\psi_{s1} : 8.0933° & X_{s2} : 252.5458 \\
\psi_{s2} : 40.5160° & Y_{s2} : -15.1531
\end{array}
\tag{13}
$$

Constraint 에 의한 또 다른 특성은 **θ_1 과 θ_2 간의 관계가 일정**하다는 것이다. 즉, θ_1 에 의해 θ_2 가 구속되어 있음을 알 수 있다. 따라서 <그림 8>과 같이 Geometric 방법으로



<그림 8> θ_1과 θ_2의 관계

θ_1 과 θ_2 사이의 관계를 구하였다. 이는 나중에 θ_1 의 값만 알고도 θ_2 의 값을 알 수 있음을 보여준다.

관계식은 다음과 같은 과정을 거쳐 (15)와 같은 관계식을 구하였다.

$$L = \sqrt{(547.5 - 185\cos\theta_1)^2 + (330 - 185\,sin\,\theta_1)^2}$$

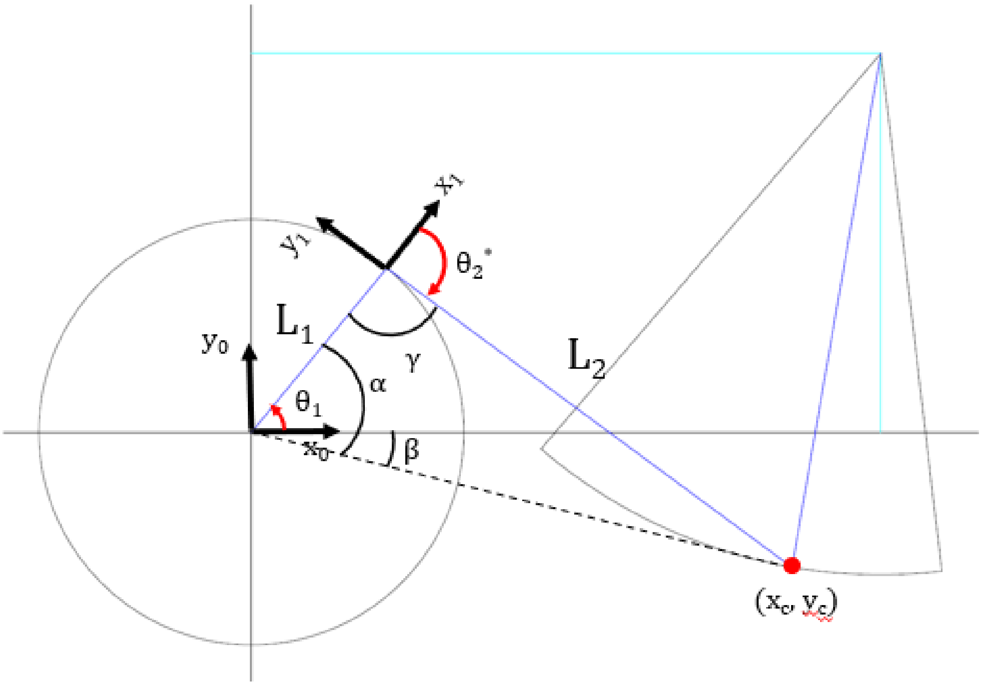
$$cos\omega = D = \frac{L^2 + {L_2}^2 - {L_4}^2}{2 * L * L_2} \tag{14}$$

$$\omega = atan2\left(\sqrt{1 - D^2}, D\right)$$

$$\tau = atan2(547.5 \ - 185cos\theta_1, 330 \ - 185sin\theta_1)$$

$$\therefore \ \ \theta_2^* = -(\omega + \tau) + 90° \ - \ \theta_1 \tag{15}$$

다음으로 역기구학을 풀기 위해 <그림 9>와 같이 도식하였으며, Constraint 에 의해 (X_c, Y_c) 위치에 따라 Upper arm 일 때와 Under arm 인 경우의 θ_1과 θ_2의 값을 구하였다.



<그림 9> I.K.

다음은 역기구학을 풀기 위한 식이며, (16), (17)과 같은 관계식을 구하였다.

$$\beta = atan2(y_c, x_c)$$

$$\cos\alpha = D_1 = \frac{{x_c}^2 + {y_c}^2 + {L_1}^2 - {L_2}^2}{2 * \sqrt{{x_c}^2 + {y_c}^2} * L_1}$$

$$\alpha = atan2\left(\pm\sqrt{1 - {D_1}^2}, D_1\right)$$

(α가 양수이면 upper arm, 음수이면 under arm)

$$\theta_1 = \alpha + \beta \tag{16}$$

$$\cos\gamma = D_2 = \frac{{L_1}^2 + {L_2}^2 - {x_c}^2 - {y_c}^2}{2 * L_1 * L_2}$$

$$\gamma = atan2\left(\pm\sqrt{1 - {D_2}^2}, D_2\right)$$

① upper arm 일 때 ($\gamma \geq 0$)

$$\theta_2 = (\gamma - 180) + 15.9233$$

② upper arm 일 때 ($\gamma \leq 0$) (17)

$$\theta_2 = (\gamma + 180) + 15.9233$$

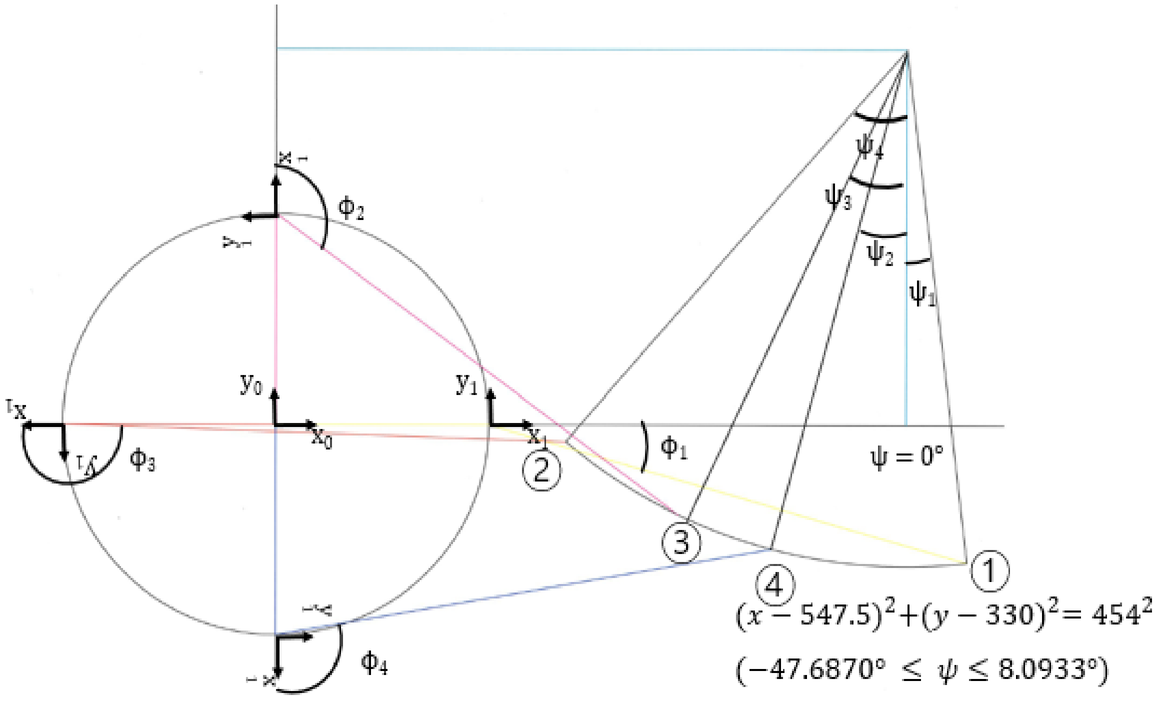
마지막으로 End Effector 의 위치인 O_3 를 구하기 위해 F.K.로 구한 T_3^0를 이용하면 다음과 같이 나오는 것을 확인할 수 있다.

$$O_3 = \begin{bmatrix} x_C + L_3 * \cos(\theta_1 + \theta_2 + \theta_3) \\ y_C + L_3 * \sin(\theta_1 + \theta_2 + \theta_3) \\ 0 \end{bmatrix} \tag{18}$$

6. 기구학 검증

기구학을 검증하기 위해 두 가지 방법을 선정하였다. 첫 번째는 **Constraint 조건을 검증**하는 것으로, Constraint에 의한 결과값들이 정확한지에 대해 진행하였고, 두 번째는 Constraint 조건에서 임의의 (X_c, Y_c)가 선정되었을 때, I.K.와 F.K.의 결과값을 비교하여, **I.K.와 F.K.를 검증**하였다.

첫 번째 방법으로 <그림 10>같은 4개의 조건(θ_1= 0°, 90°, 180°, -90°)에서 I.K.를 통해 구한(위에서 Constraint을 통해 구한) (x, y)좌표와 F.K.를 통해 구한(x, y)좌표가 동일한지 비교를 해보았다.



<그림 10> Constraint 조건 검증

1번 조건은 θ_1= 0°, θ_2^* = -15.9233°(θ_2 =0°)일 때이며, I.K. 값은 (606.1939, -120.1654) 이었으며, θ_1, θ_2 값을 F.K.에 대입하여 구한 값은(606.1939, -120.1654)로 두 결과값이 같음을 확인할 수 있었다.

```
th1 =
    90
th2star =
 -127.0106
th2 =
 -111.0873
A1 =
     0    -1     0     0
     1     0     0   185
     0     0     1     0
     0     0     0     1
A2 =
   -0.6020    0.7985         0 -263.6597
   -0.7985   -0.6020         0 -349.7536
         0         0    1.0000         0
         0         0         0    1.0000
T2 =
    0.7985    0.6020         0  349.7536
   -0.6020    0.7985         0  -78.6597
         0         0    1.0000         0
         0         0         0    1.0000
```

<그림 11> Constraint 조건 검증 결과 1

2번 조건은 θ_1 = 90°, θ_2^* = -127.0106°(θ_2 =-111.0873)일 때이며, I.K. 값은 (349.7536, -78.6597) 이었으며, θ_1, θ_2 값을 F.K.에 대입하여 구한 값은(349.7536, -78.6597)로 두 결과값이 같음을 확인할 수 있었다.

```
th1 =
     0
th2star =
  -15.9233
th2 =
     0
A1 =
     1     0     0   185
     0     1     0     0
     0     0     1     0
     0     0     0     1
A2 =
    0.9616    0.2744         0  421.1939
   -0.2744    0.9616         0 -120.1654
         0         0    1.0000         0
         0         0         0    1.0000
T2 =
    0.9616    0.2744         0  606.1939
   -0.2744    0.9616         0 -120.1654
         0         0    1.0000         0
         0         0         0    1.0000
```

<그림 12> Constraint 조건 검증 결과 2

3번 조건은 θ_1 = 180°, θ_2^* = 177.9927°(θ_2 =-166.0840)일 때이며, I.K. 값은 (252.7312, -15.3418) 이었으며, θ_1, θ_2 값을 F.K.에 대입하여 구한 값은(252.7312, -15.3418)로 두 결과값이 같음을 확인할 수 있었다.

```
th1 =
   180
th2star =
  177.9927
th2 =
 -166.0840
A1 =
    -1     0     0  -185
     0    -1     0     0
     0     0     1     0
     0     0     0     1
A2 =
   -0.9994   -0.0350         0 -437.7312
    0.0350   -0.9994         0   15.3418
         0         0    1.0000         0
         0         0         0    1.0000
T2 =
    0.9994    0.0350         0  252.7312
   -0.0350    0.9994         0  -15.3418
         0         0    1.0000         0
         0         0         0    1.0000
```

<그림 13> Constraint 조건 검증 결과 3

마지막으로 4번 조건은 θ_1 = -90°, θ_2^* = 100.0099°(θ_2 = 115.9332)일 때이며, I.K. 값은 (431.3326, -108.8676) 이었으며, θ_1, θ_2 값을 F.K.에 대입하여 구한 값은(431.3326, -108.8676)로 두 결과값이 같음을 확인할 수 있었다

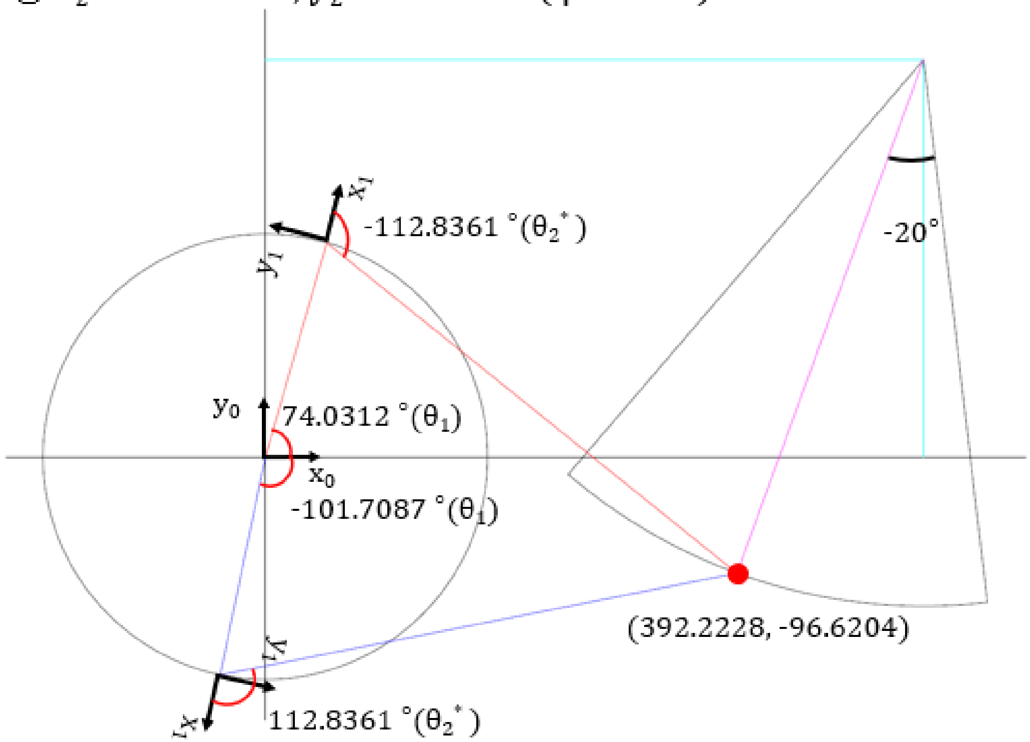
```
th1 =
   -90
th2star =
  100.0099
th2 =
  115.9332
A1 =
     0     1     0     0
    -1     0     0  -185
     0     0     1     0
     0     0     0     1
A2 =
   -0.1738   -0.9848         0  -76.1324
    0.9848   -0.1738         0  431.3326
         0         0    1.0000         0
         0         0         0    1.0000
T2 =
    0.9848   -0.1738         0  431.3326
    0.1738    0.9848         0 -108.8676
         0         0    1.0000         0
         0         0         0    1.0000
```

<그림 14> Constraint 조건 검증 결과 4

이로써 첫 번째 검증으로 Constraint에 의한 조건을 검증할 수 있었다.

두 번째 검증으로 Constraint에 의한 임의의 (X_c, Y_c)를 선정했을 때, I.K.로 풀어서 나온 θ_1, θ_2 값을 F.K.에 넣었을 때, 동일한 (X_c, Y_c)가 나오는지 검증을 진행하였다.

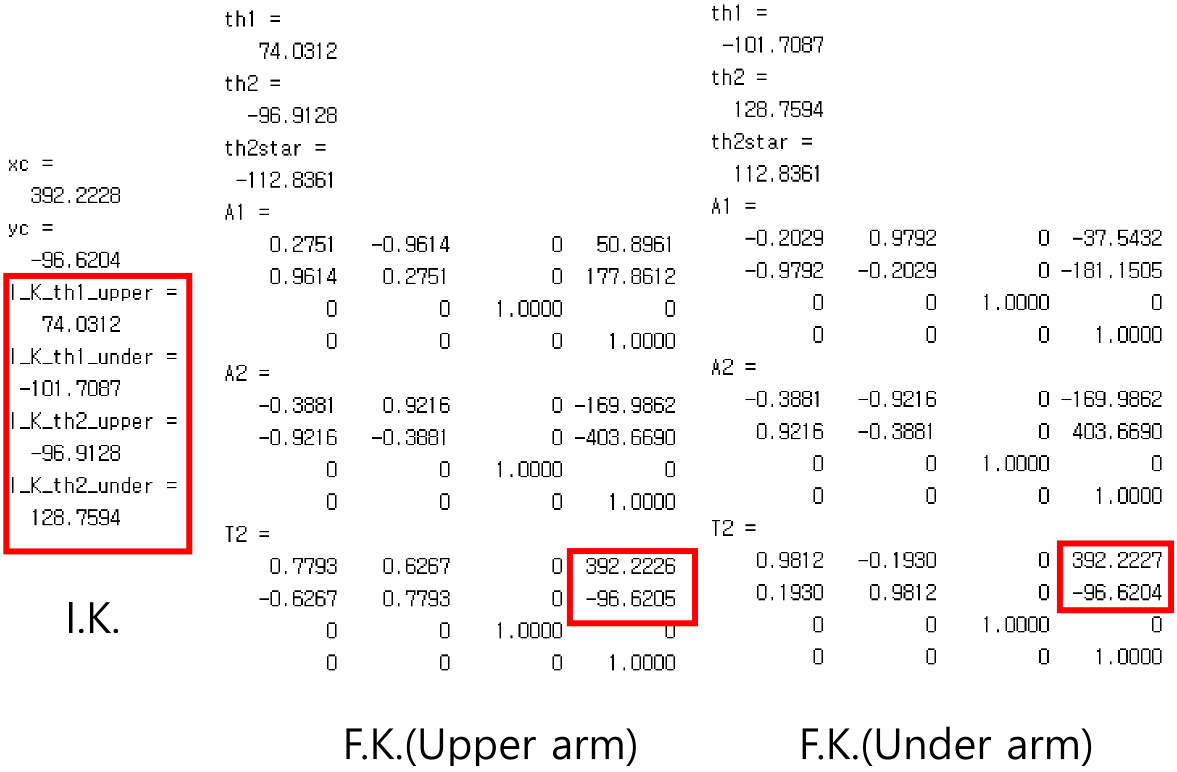
① $x_2 = 392.2228, y_2 = -96.6204(\psi = -20°)$



<그림 15> F.K. 및 I.K. 검증 1

<그림 15>와 같이 ψ가 -20°일 때 (X_c, Y_c)는 (392.2228, -96.6204)가 나옴을 알 수 있다. 이 때, I.K.를 풀었을 때 나오는 값은 Upper arm일 때 θ_1= 74.0312°, θ_2 = -96.9128°이며, 이 값을 F.K.에 넣었을 때 나오는 값이 X_c = 392.2226, Y_c = -96.6205 가 나왔다. 이 값은 임의로 선정한 (X_c, Y_c)값과 거의 비슷함을 알 수 있다.

반대로 Under arm일 때 θ_1 = -101.7087°, θ_2 = 128.7594°이며, 이 값을 F.K.에 넣었을 때 나오는 값이 X_c = 392.2227, Y_c = -96.6204 가 나왔다. 이 값은 임의로 선정한 (X_c, Y_c)값과 동일함을 알 수 있다. 이 결과 통해 Upper arm과 Under arm 모두 F.K.와 I.K.가 적용됨을 검증할 수 있다.



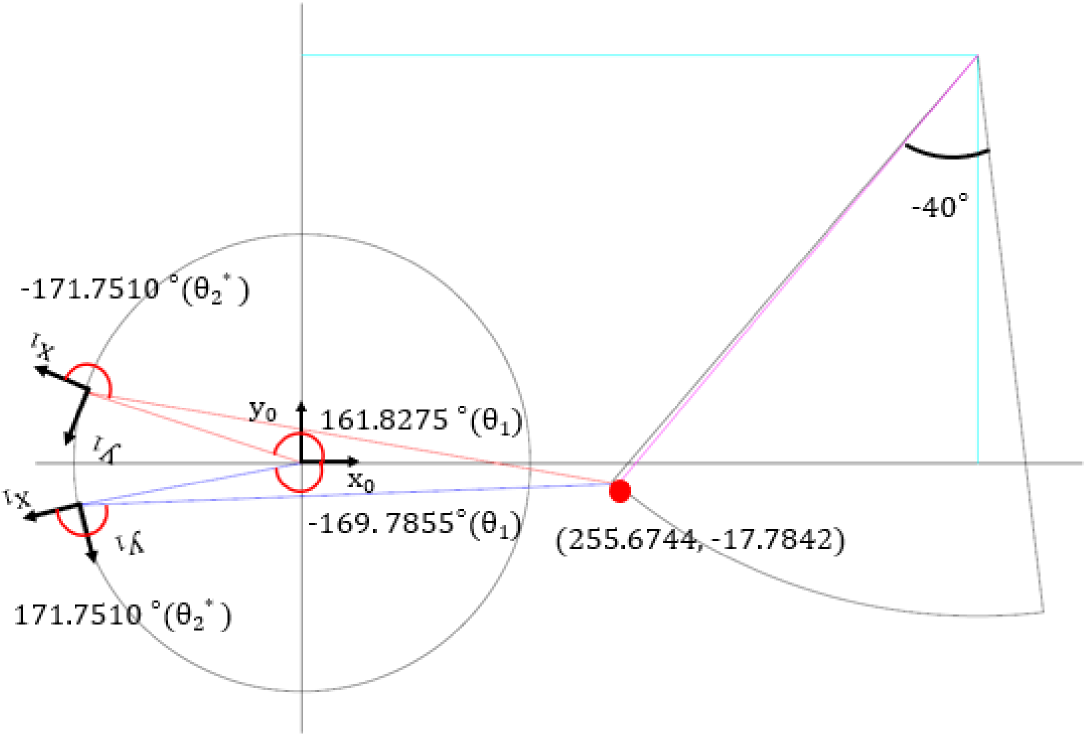
<그림 16> F.K. 및 I.K. 검증 결과 1

이와 같은 과정을 한번 더 진행하였는데, 그 이유는 θ_1 값이 90°이상일 때와 -90°이하일 때의 값도 검증하기 위함이다.

이번에는 ψ가 -40°일 때 (X_c, Y_c)는 (255.6744, -17.7842)가 나옴을 알 수 있다. 이 때, I.K.를 풀었을 때 나오는 값은 Upper arm일 때 θ_1 = 161.8275°, θ_2 = -155.8277°이며, 이 값을 F.K.에 넣었을 때 나오는 값이 X_c = 255.6744, Y_c = -17.7842가 나왔다. 이 값은 임의로 선정한 (X_c, Y_c)값과 동일함을 알 수 있다.

반대로 Under arm일 때 θ_1 = -169.7855°, θ_2 = -172.3257°이며, 이 값을 F.K.에 넣었을 때 나오는 값이 X_c = 255.6744, Y_c = -17.7844 가 나왔다. 이 값은 임의로 선정한 (X_c, Y_c)값과 동일함을 알 수 있다. 이를 통해 모든 θ_1 값에 대해서 F.K.와 I.K.가 검증됨을 보일 수 있다.

② $x_2 = 255.6744,\ y_2 = -17.7842(\psi = -40°)$



<그림 17> F.K. 및 I.K. 검증 2

```
xc =
  255.6744
yc =
  -17.7842
I_K_th1_upper =
  161.8275
I_K_th1_under =
 -169.7855
I_K_th2_upper =
 -155.8277
I_K_th2_under =
 -172.3257
```

I.K.

```
th1 =
  161.8275
th2 =
 -155.8277
th2star =
 -171.7510
A1 =
   -0.9501   -0.3119         0 -175.7725
    0.3119   -0.9501         0   57.6976
         0         0    1.0000         0
         0         0         0    1.0000
A2 =
   -0.9897    0.1435         0 -433.4684
   -0.1435   -0.9897         0  -62.8422
         0         0    1.0000         0
         0         0         0    1.0000
T2 =
    0.9850    0.1723         0  255.6744
   -0.1723    0.9850         0  -17.7843
         0         0    1.0000         0
         0         0         0    1.0000
```

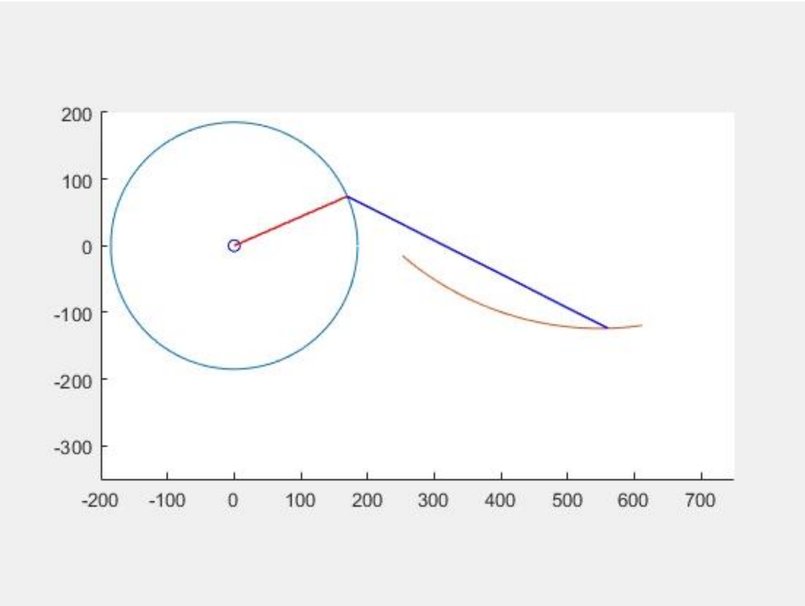
F.K.(Upper arm)

```
th1 =
 -169.7855
th2 =
 -172.3257
th2star =
  171.7510
A1 =
   -0.9842    0.1773         0 -182.0679
   -0.1773   -0.9842         0  -32.8068
         0         0    1.0000         0
         0         0         0    1.0000
A2 =
   -0.9897   -0.1435         0 -433.4684
    0.1435   -0.9897         0   62.8422
         0         0    1.0000         0
         0         0         0    1.0000
T2 =
    0.9994   -0.0343         0  255.6744
    0.0343    0.9994         0  -17.7844
         0         0    1.0000         0
         0         0         0    1.0000
```

F.K.(Under arm)

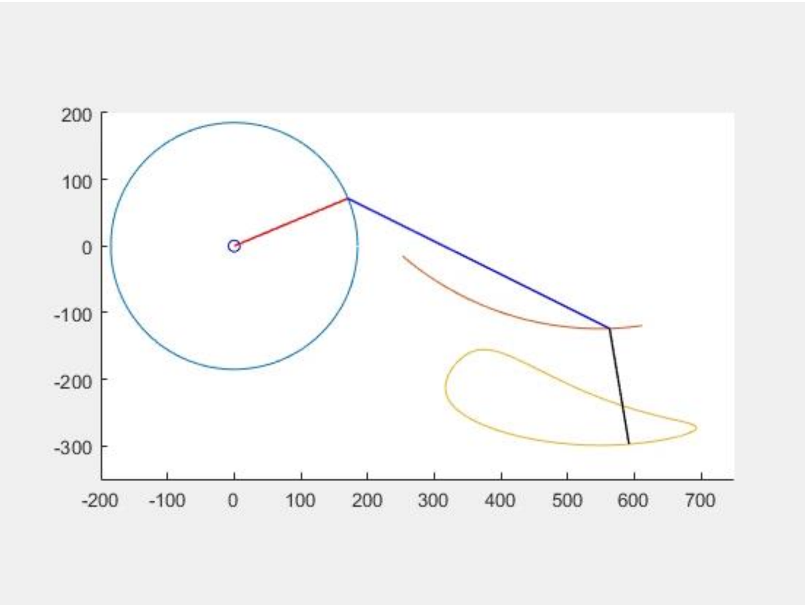
<그림 18> F.K. 및 I.K. 검증 결과

두 번째 검증방법 원리를 이용해 Constraint 에 의한 점들을 나열하여, I.K.로 풀고, 이를 F.K.로 각각의 점들을 구하여 MATLAB 으로 **애니메이션**을 구하면 <그림 19>와 같이 나온다.



<그림 19> 검증 결과 Animation

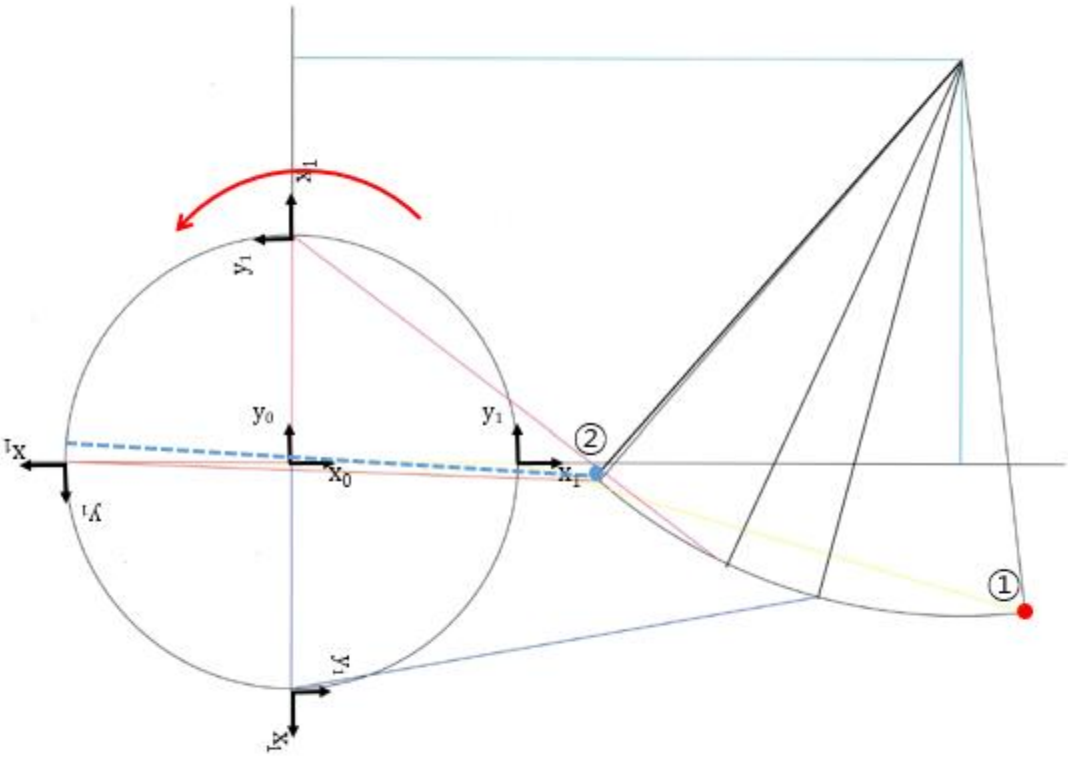
그리고, End Effector 의 위치까지 고려하면 <그림 20>과 같이 나오며, 이는 **실제 로봇의 보행궤적**과 동일하다.



<그림 20> 검증 결과 Animation 2

7. 경로계획

경로계획을 세우고자 할 때, 앞의 애니메이션에서 보았듯이, End Effector의 경로를 고려하기에는 그 형태가 너무 불규칙하여 원의 방정식에 구속되어 있는 **O_2의 경로계획**을 세웠다. O_2의 경로는 다음 그림과 같이 원의 방정식 일부분이며, ①번 위치에서 ②번 위치까지 선정하였다. ①번 위치는 θ_1이 0°인 위치이며, ②번 위치는 Singularity에 걸린 위치로, Upper arm에서 Under arm으로 넘어가는 순간이다. 이렇게 경로를 선정한 이유는 먼저 **Upper arm일 때의 구간을 해석**하기 위함으로, 나중에는 Under arm에 대해서 추가적으로 연구가 필요하다.



<그림 21> 경로계획

O_2의 Trajectory Planning을 계획하기 위해 Cubic Polynomial Trajectories를 사용하였으며, 운동 시간은 1초로 선정하였다. 1초로 선정한 이유는 현재 로봇이 2km/h 로 운동하기 위해서는 왕복운동 했을 때 대략 1.8초가 나오기 때문이다.

Cubic Polynomial Trajectories를 풀기 위해 다음과 같이 경계조건을 주었다.

초기위치 ① (606.1939, -120.1654) / $t_o = 0,\ v_0 = 0$ (19)

Final position ② (252.5458, -15.1531) / $t_f = 1,\ v_f = 0$ (20)

이 조건을 이용하여 x좌표의 Trajectory를 구하면 다음과 같이 나온다.

$$q_x(t) = 606.1939 - 1060.9443\ *\ t^2 + 707.2962\ *\ t^3 \tag{21}$$

이 때 y 좌표의 Trajectory 는 원의 방정식($(x - 547.5)^2 + (y - 330)^2 = 454^2$)에 구속되어 있기 때문에 x 좌표를 원의 방정식에 넣을 때 다음과 같이 y 좌표의 trajectory 를 구할 수 있었다.

$$q_y(t) = 330 - \sqrt{206116 - (58.6939 - 1060.94t^2 + 707.296t^3)^2} \tag{22}$$

이를 각각 미분하여, 속도와 가속도 식을 구하면 다음과 같다.

$$\dot{q}_x(t) = -2121.89\ * t + 2121.89\ *\ t^2 \tag{23}$$

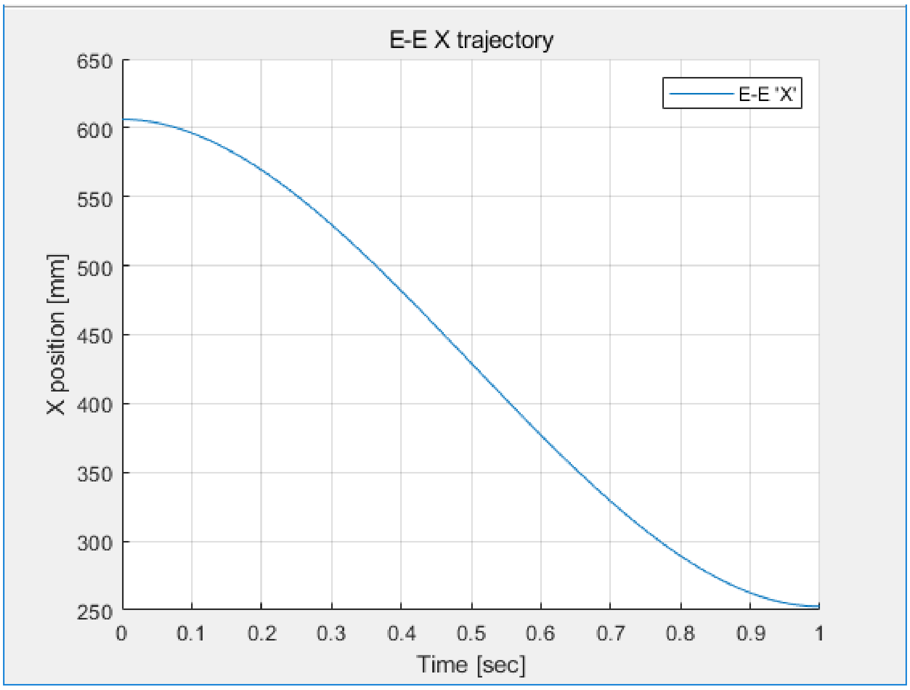
$$\ddot{q}_x(t) = -4330.89 + 4243.78\ * t \tag{24}$$

$$\dot{q}_y(t) = \frac{(-2121.89t + 2121.89t^2)(58.6939 - 1060.94t^2 + 707.296t^3)}{\sqrt{206116 - (58.6939 - 1060.94t^2 + 707.296t^3)^2}} \tag{25}$$

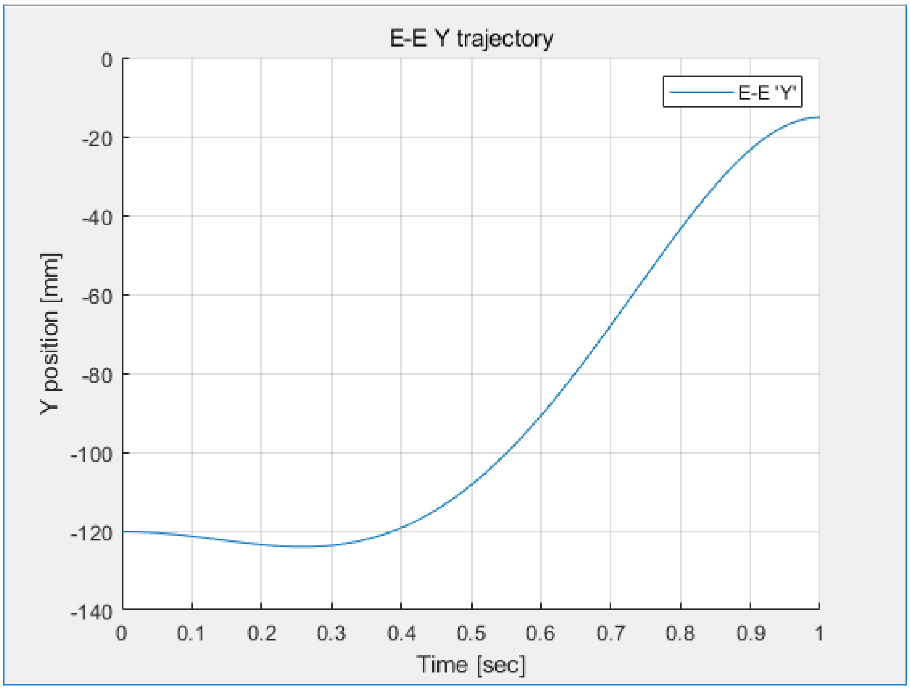
$$\ddot{q}_y(t) = \frac{(-2121.89t + 2121.89t^2)^2(58.6939 - 1060.94t^2 + 707.296t^3)^2}{(206116 - (58.6939 - 1060.94t^2 + 707.296t^3)^2))^{3/2}} \tag{26}$$

$$+ \frac{(-2121.89t + 2121.89t^2)^2}{\sqrt{206116 - (58.6939 - 1060.94t^2 + 707.296t^3)^2}} + \frac{(-2121.89 + 4243.78t)(58.6939 - 1060.94t^2 + 707.296t^3)}{\sqrt{206116 - (58.6939 - 1060.94t^2 + 707.296t^3)^2}} \tag{27}$$

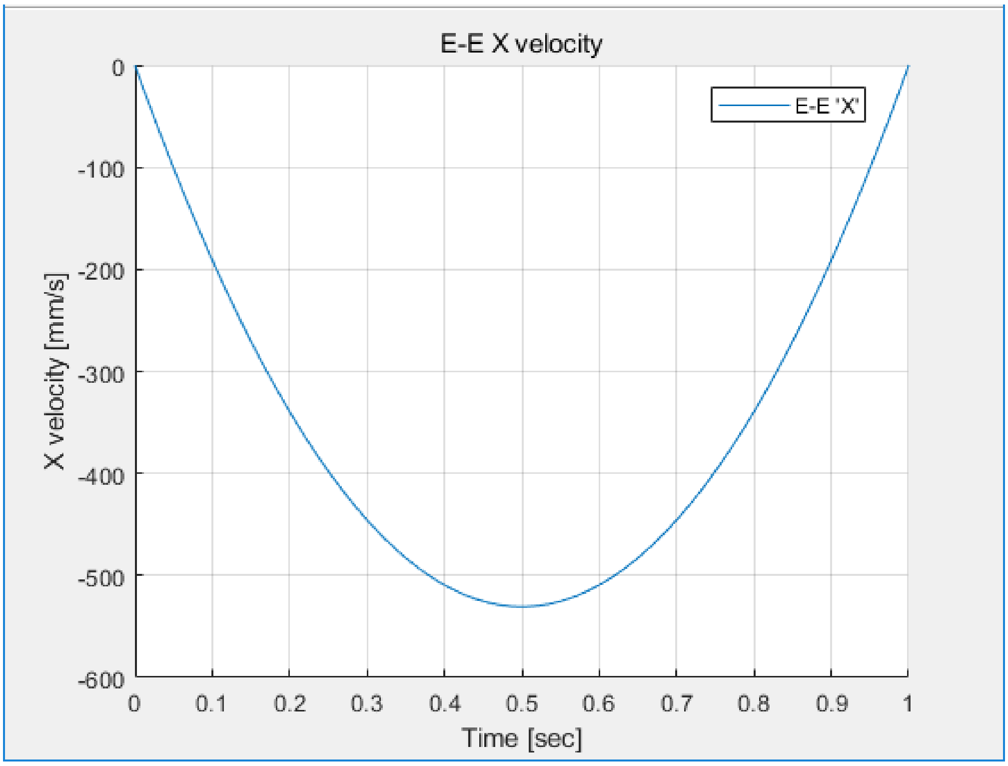
이렇게 구한 식을 시간에 대한 그래프로 나타내면 다음과 같이 그릴 수 있다.



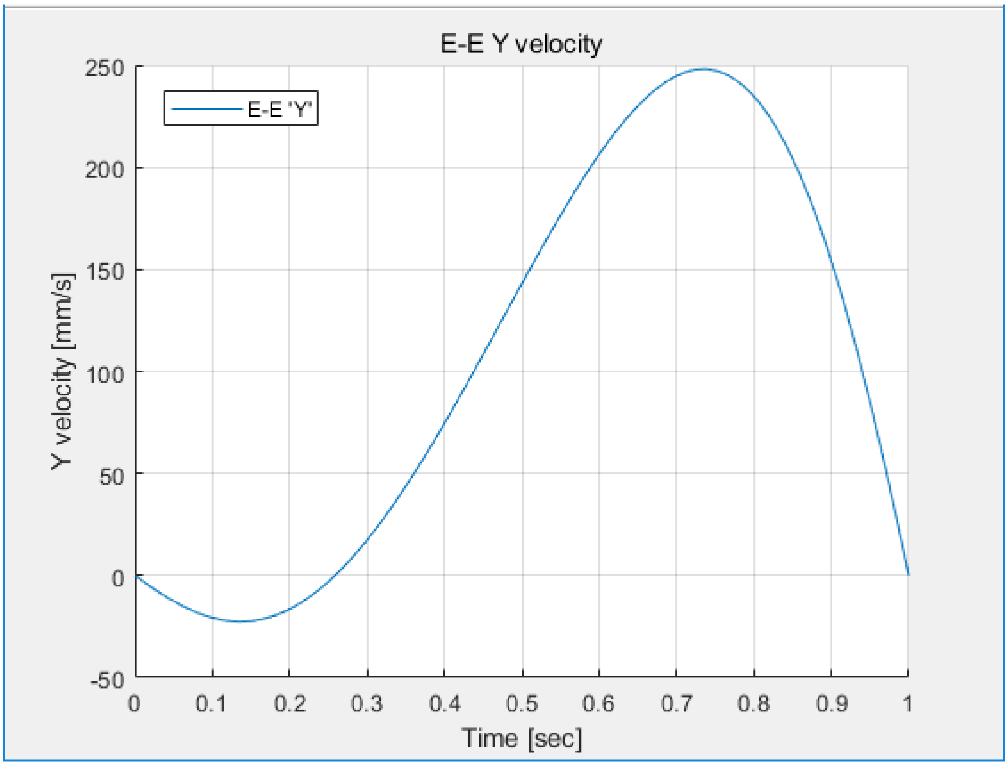
<그림 22> X Trajectory



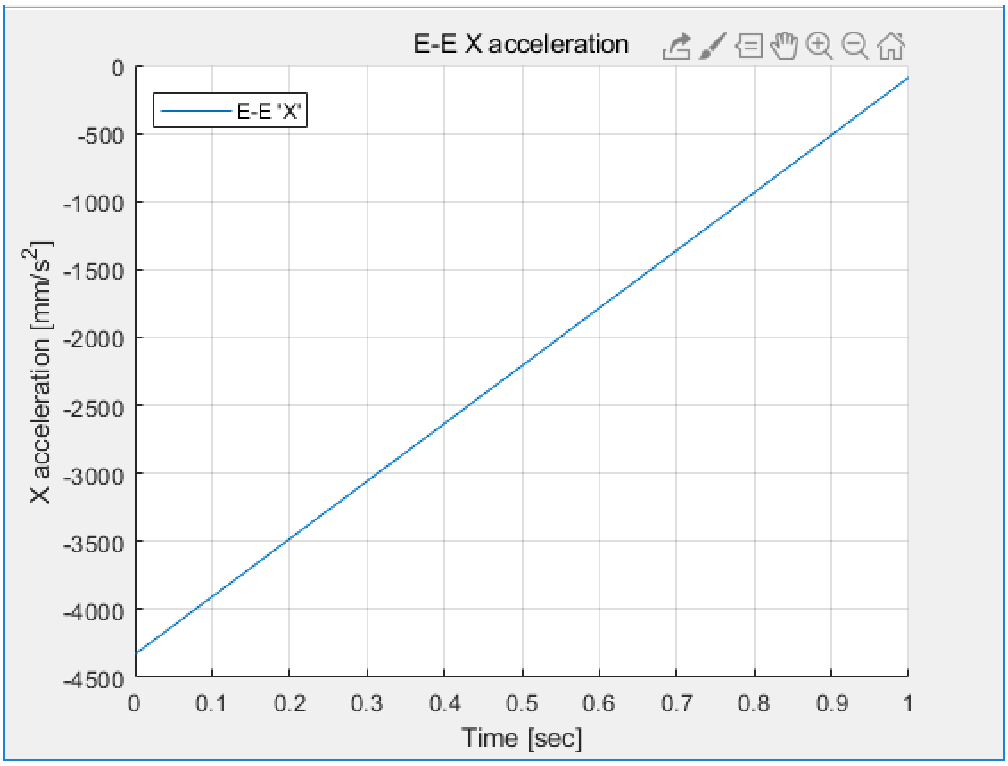
<그림 23> Y Trajectory



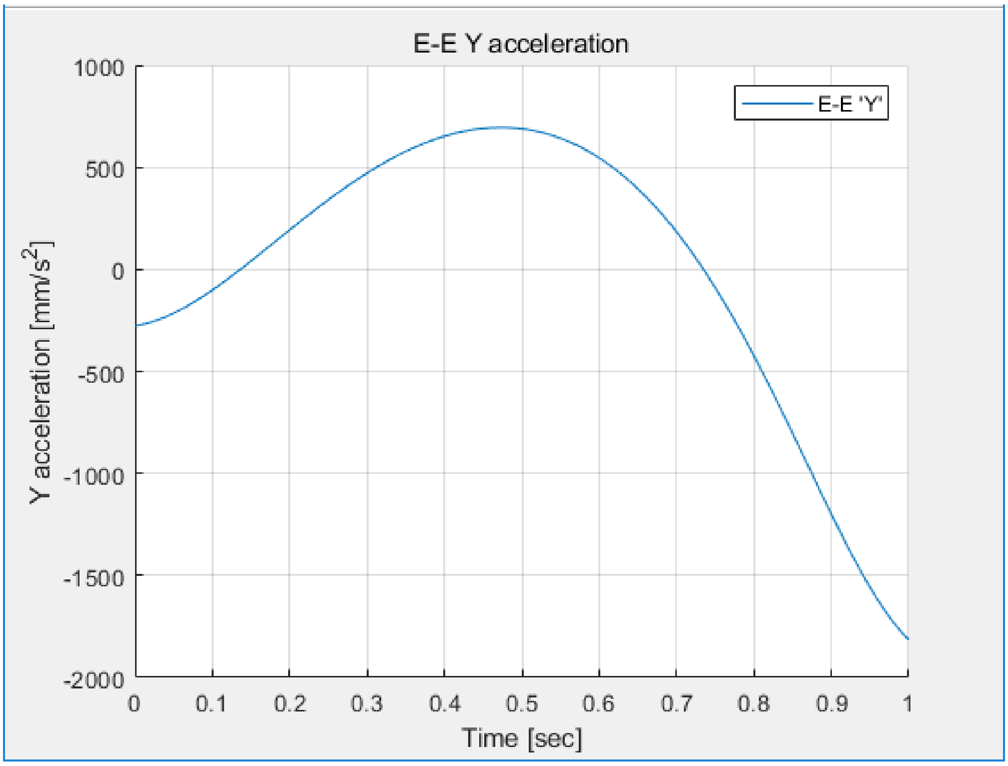
<그림 24> X Velocity



<그림 25> Y Velocity

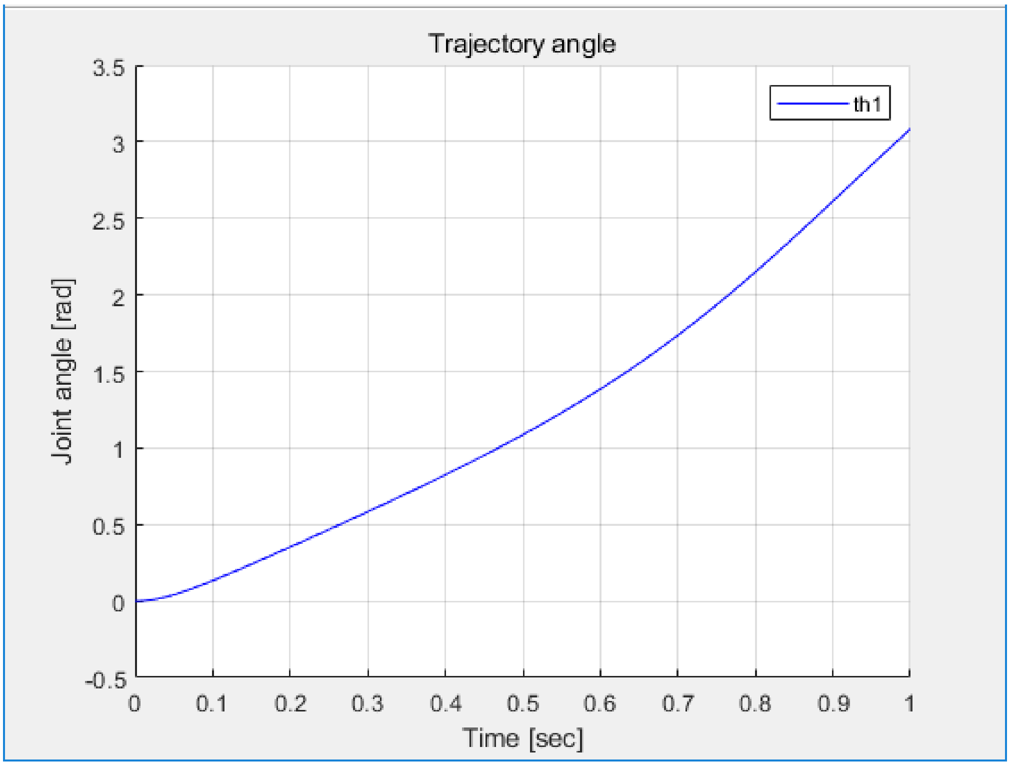


<그림 26> X Acceleration

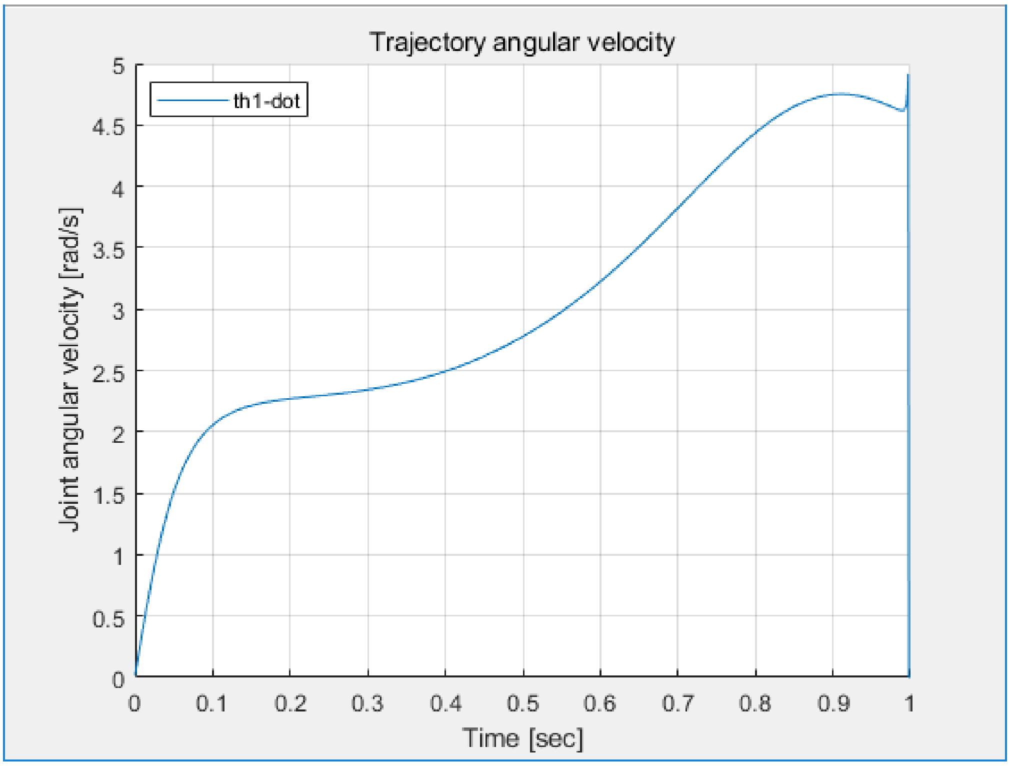


<그림 27> Y Acceleration

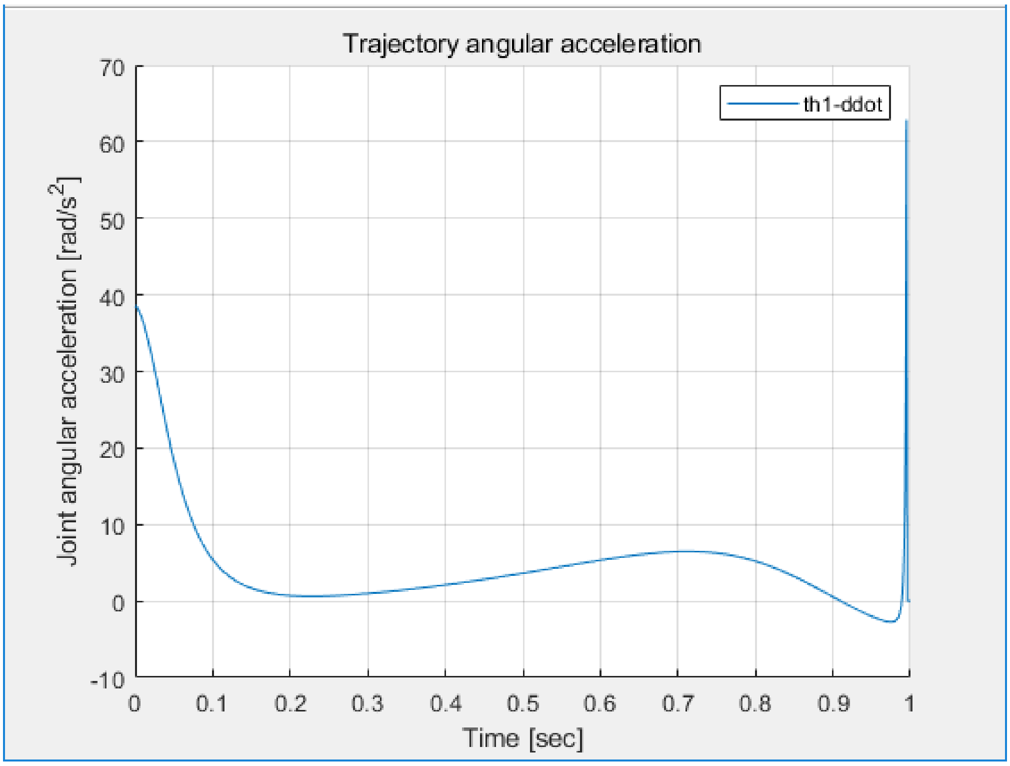
이렇게 구한 x와 y 좌표의 Trajectory를 이용하여 I.K.를 풀면 각각 θ_1과 θ_2를 구할 수 있다. 여기서 **Joint variable은 θ_1 하나**이기 때문에 θ_1의 Trajectory와 각속도, 각가속도를 각각 그래프로 나타내었다. 이때, 각속도와 각가속도는 미분할 수 없으므로, θ_1의 값을 차분하여 만들어 내었다. 그런데, 차분할 때 값이 튀는 경우가 발생하였는데, 이 경우에 그 값을 버리고 값을 선정하였다.



<그림 28> θ_1 Trajectory



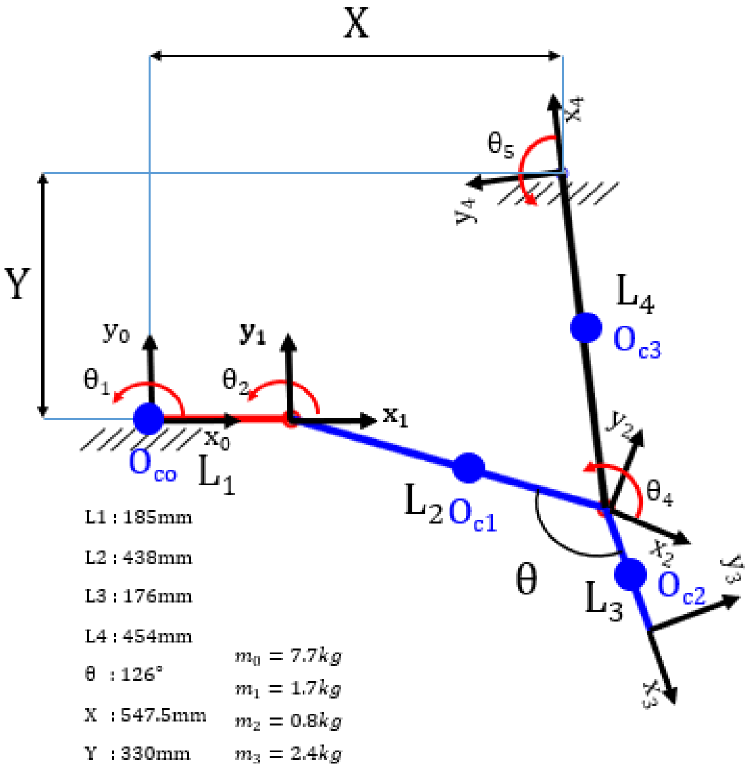
<그림 29> θ_1 Angular Velocity



<그림 30> θ_1 Angular Acceleration

8. 동역학 해석

동역학 해석을 하기 위해 다음과 같이 **각 링크의 질량중심**을 구하였다. 여기서 링크2와 링크3은 두 개로 나누어 진행하였으며, 그 무게는 각각 1.7kg, 0.8kg으로 나누었다.



<그림 31> 각 링크의 질량 중심

각각의 질량중심은 다음과 같이 구하였다.

$$O_{co} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{28}$$

$$O_{c1} = \frac{O_2 + O_1}{2} = \begin{bmatrix} L_1 \cos\theta_1 + \dfrac{L_2 \cos(\theta_1+\theta_2)}{2} \\ L_1 \sin\theta_1 + \dfrac{L_2 \sin(\theta_1+\theta_2)}{2} \\ 0 \end{bmatrix} \tag{29}$$

$$O_{c2} = \frac{O_3 + O_2}{2} = \begin{bmatrix} L_1 \cos\theta_1 + L_2\cos(\theta_1+\theta_2) + \dfrac{L_3 \cos(\theta_1+\theta_2+\theta_3)}{2} \\ L_1 \sin\theta_1 + L_2 \sin(\theta_1+\theta_2) + \dfrac{L_3 \sin(\theta_1+\theta_2+\theta_3)}{2} \\ 0 \end{bmatrix} \tag{30}$$

$$O_{c3} = \frac{O_4 + O_2}{2} = \begin{bmatrix} \dfrac{X + L_1 \cos\theta_1 + L_2 \cos(\theta_1+\theta_2)}{2} \\ \dfrac{Y + L_1 \sin\theta_1 + L_2 \sin(\theta_1+\theta_2)}{2} \\ 0 \end{bmatrix} \tag{31}$$

먼저 **Kinetic Energy(K)**를 구하기 위해 Translational Kinetic Energy(K_T)와 Rotational Kinetic Energy(K_R)를 구하였다. K_T는 다음과 같다.

$$K_T = \frac{1}{2}\dot{q}^T\left(m_1 J_{v_{c1}}^T J_{v_{c1}} + m_2 J_{v_{c2}}^T J_{v_{c2}} + m_3 J_{v_{c3}}^T J_{v_{c3}}\right)\dot{q} \tag{32}$$

여기서 Jacobian 이 필요하므로, 각각의 Jacobian 을 다음과 같이 구하였다.

$$J_{v_{c1}} = \begin{bmatrix} -L_1 \sin\theta_1 - \dfrac{L_2 \sin(\theta_1+\theta_2)}{2} \\ L_1\cos\theta_1 + \dfrac{L_2\cos(\theta_1+\theta_2)}{2} \\ 0 \end{bmatrix} \tag{32}$$

$$J_{v_{c2}} = \begin{bmatrix} -L_1 \sin\theta_1 - L_2\sin(\theta_1+\theta_2) - \dfrac{L_3 \sin(\theta_1+\theta_2+\theta_3)}{2} \\ L_1\cos\theta_1 + L_2\cos(\theta_1+\theta_2) + \dfrac{L_3\cos(\theta_1+\theta_2+\theta_3)}{2} \\ 0 \end{bmatrix} \tag{33}$$

$$J_{v_{c3}} = \begin{bmatrix} -\dfrac{Y + L_1\sin\theta_1 + L_2\sin(\theta_1+\theta_2)}{2} \\ \dfrac{X + L_1\cos\theta_1 + L_2\cos(\theta_1+\theta_2)}{2} \\ 0 \end{bmatrix} \tag{34}$$

K_R은 다음과 같다.

$$\begin{aligned} K_R = \frac{1}{2}\dot{q}^T\big(&J_{\omega_{c0}}^T R_0^0 I_{c1}(R_0^0)^T J_{\omega_{c0}} + J_{\omega_{c1}}^T R_{c1}^0 I_{c1}(R_{c1}^0)^T J_{\omega_{c1}} + J_{\omega_{c2}}^T R_{c2}^0 I_{c2}(R_{c2}^0)^T J_{\omega_{c2}} \\ &+ J_{\omega_{c3}}^T R_{c3}^0 I_{c3}(R_{c3}^0)^T J_{\omega_{c3}}\big)\dot{q} \end{aligned} \tag{35}$$

여기서도 마찬가지로 Jacobian 이 필요하므로 다음과 같이 구하였다.

$$J_{\omega_{c0}} = J_{\omega_{c1}} = J_{\omega_{c2}} = J_{\omega_{c2}} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{36}$$

여기서 각속도에 의한 Jacobian 은 Planar 로봇으로 모두 위와 같이 동일하게 나온다.

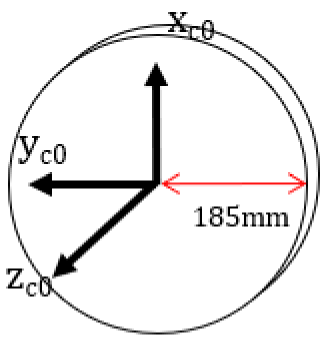
그리고 F.K.에서 구한 Homogeneous Vector 에서 Rotation Vector 를 구하면 다음과 같다.

$$R_{c1}^0 = R_2^0 = \begin{bmatrix} \cos(\theta_1+\theta_2^*) & -\sin(\theta_1+\theta_2^*) & 0 \\ \sin(\theta_1+\theta_2^*) & \cos(\theta_1+\theta_2^*) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{37}$$

$$R_{c2}^{0} = R_{3}^{0} = \begin{bmatrix} \cos(\theta_1 + \theta_2^* + \theta_3) & -\sin(\theta_1 + \theta_2^* + \theta_3) & 0 \\ \sin(\theta_1 + \theta_2^* + \theta_3) & \cos(\theta_1 + \theta_2^* + \theta_3) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{38}$$

$$R_{c3}^{0} = R_{4}^{0} = \begin{bmatrix} \cos(\theta_1 + \theta_2^* + \theta_4) & -\sin(\theta_1 + \theta_2^* + \theta_4) & 0 \\ \sin(\theta_1 + \theta_2^* + \theta_4) & \cos(\theta_1 + \theta_2^* + \theta_4) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{39}$$

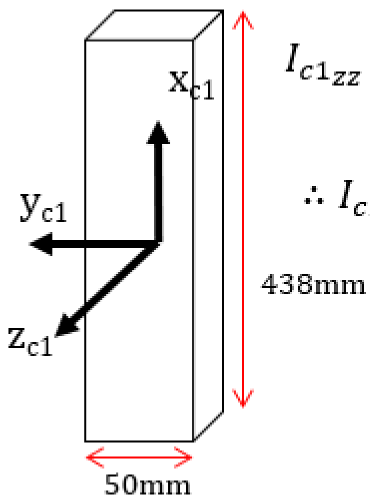
또한 각 링크의 Inertial Tensor 를 구해야 하는데, 여기서는 z 축의 Planar 로봇의 특징으로 질량 관성 모멘트만 구하면 되므로, 다음과 같이 구하였다.



$$I_{c0_{zz}} = \frac{m_0}{2} R^2$$

$$\therefore I_{c0_{zz}} = 0.1318\, kg \cdot m^2$$

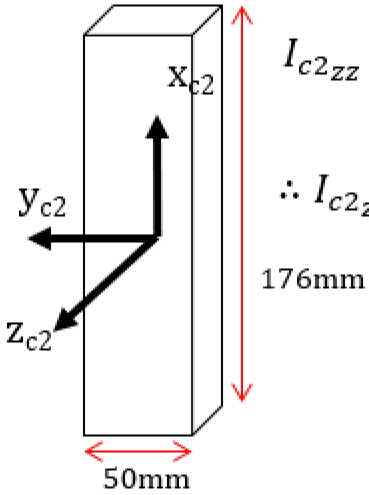
<그림 32> 1번 링크 질량 관성모멘트



$$I_{c1_{zz}} = \frac{m_1}{12}\left({L_{xc1}}^2 + {L_{yc1}}^2\right)$$

$$\therefore I_{c2_{zz}} = 0.0275\, kg \cdot m^2$$

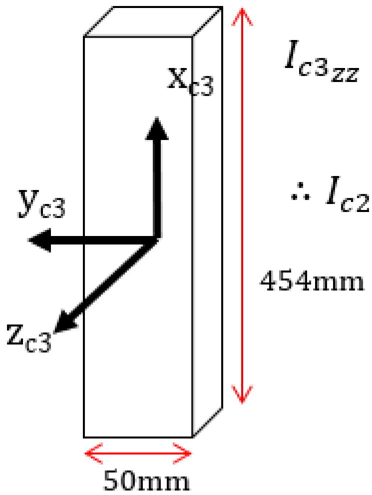
<그림 33> 2번 링크 질량 관성모멘트



$$I_{c2_{zz}} = \frac{m_2}{12}\left({L_{xc2}}^2 + {L_{yc2}}^2\right)$$

$$\therefore I_{c2_{zz}} = 0.002223\ kg \cdot m^2$$

<그림 34> 3번 링크 질량 관성모멘트



$$I_{c3_{zz}} = \frac{m_1}{12}\left({L_{xc3}}^2 + {L_{yc3}}^2\right)$$

$$\therefore I_{c2_{zz}} = 0.04172\ kg \cdot m^2$$

<그림 35> 4번 링크 질량 관성모멘트

이와 같은 결과로 Kinetic Energy(K)를 구하면 다음과 같으며,

$$K = K_T + K_R = \frac{1}{2}\dot{q}^T D(q)\dot{q} \tag{40}$$

여기서 D(q)를 구하면 다음과 같이 구할 수 있다.

$$D(q) = d_{11} = \frac{1}{4}[4I_{c0zz} + 4I_{c1zz} + 4I_{c2zz} + 4I_{c3zz} + {L_3}^2 m_2 + {L_2}^2(m_1 + 4m_2 + m_3) + {L_1}^2(4(m_1 + m_2) + m_3) + m_3(x^2 + y^2) + 2\{L_1 m_3 x cos\theta_1 + L_1 L_2(2m_1 + 4m_2 + m_3)\cos\theta_2 + L_2 m_3 x cos(\theta_1 + \theta_2) + 2L_2 L_3 m_2 cos\theta_3 + 2L_1 L_3 m_2 \cos(\theta_2 + \theta_3) + L_1 m_3 y sin\theta_1 + L_2 m_3 y sin(\theta_1 + \theta_3)\}] \tag{41}$$

여기서 특징은 Joint variable 이 한 개이기 때문에 Vector 가 아닌 **Scalar** 로 나오게 된다.

다음으로 **Potential Energy(P)**의 식은 다음과 같다.

$$P = m_1 \cdot g \cdot y_{c1} + m_2 \cdot g \cdot y_{c2} + m_3 \cdot g \cdot y_{c3} \tag{42}$$

위 식을 구하기 위해 각각의 y 값을 구하면 다음과 같다.

$$y_{c1} = L_1 \sin\theta_1 + \frac{L_2 \sin(\theta_1 + \theta_2)}{2} \tag{43}$$

$$y_{c2} = L_1 \sin\theta_1 + L_2 \sin(\theta_1 + \theta_2) + \frac{L_3 \sin(\theta_1 + \theta_2 + \theta_3)}{2} \tag{44}$$

$$y_{c3} = \frac{Y + L_1 \sin\theta_1 + L_2 \sin(\theta_1 + \theta_2)}{2}$$

따라서 P 를 구할 수 있으며, 이 때 g 값을 구할 수 있다.

$$P = \frac{1}{2} g\{m_3 y + L_1(2(m_1 + m_2) + m_3) sin\theta_1 + L_2(m_1 + 2m_2 + m_3)\sin(\theta_1 + \theta_2) + L_3 m_2 \sin(\theta_1 + \theta_2 + \theta_3)\} \tag{46}$$

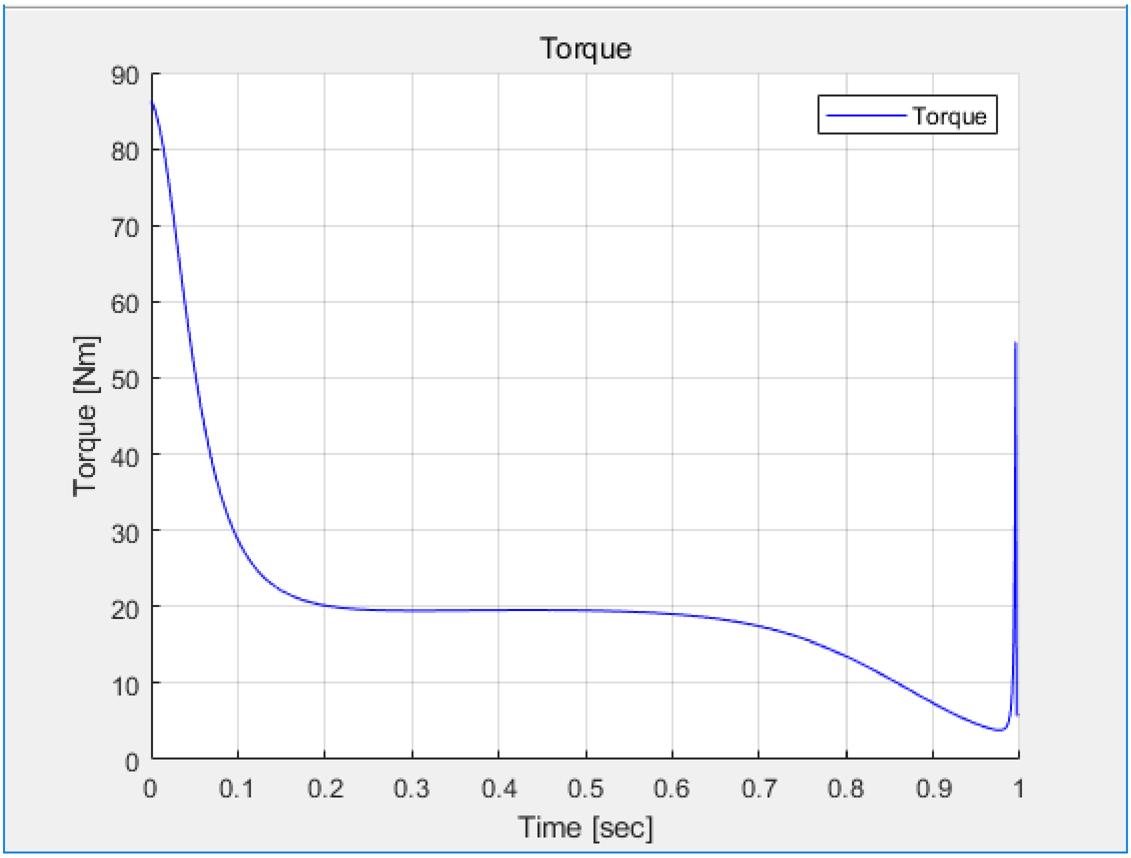
$$g = \frac{\partial P}{\partial \theta_1} = \frac{1}{2} g\{L_1(2(m_1 + m_2) + m_3) cos\theta_1 + L_2(m_1 + 2m_2 + m_3)\cos(\theta_1 + \theta_2) + L_3 m_2 \cos(\theta_1 + \theta_2 + \theta_3)\} \tag{47}$$

마지막으로 **Centrifugal and Coliolis(C)**를 구하면 다음과 같다.

$$c_{11} = c_{111} = \frac{1}{2}\frac{\partial d_{11}}{\partial q_1} = \frac{1}{4} m_3\{L_1 y cos\theta_1 + L_2 y cos(\theta_1 + \theta_2) - x(L_1 sin\theta_1 + L_2 \sin(\theta_1 + \theta_2))\} \tag{48}$$

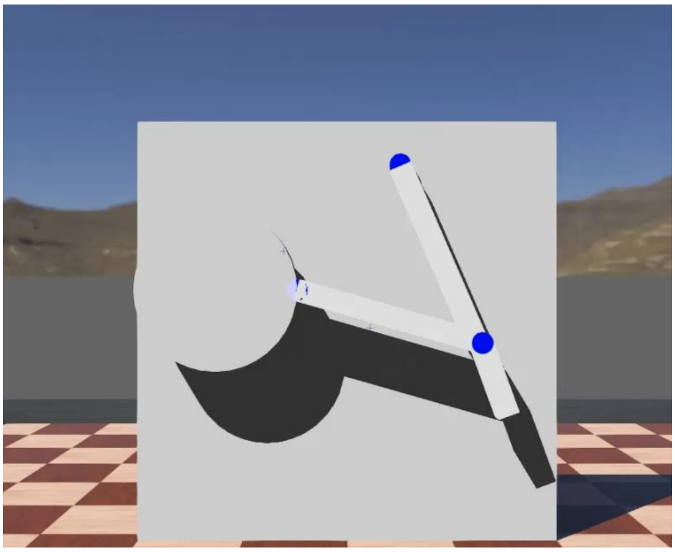
이 결과값들을 이용하여 Torque 를 계산하면 다음과 같은 그래프를 얻을 수 있다.

$$\tau = D(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = d_{11}\ddot{q} + c_{11}\dot{q} + g \tag{49}$$



<그림 36> θ_1의 Torque 그래프

9. 독립관절 위치제어

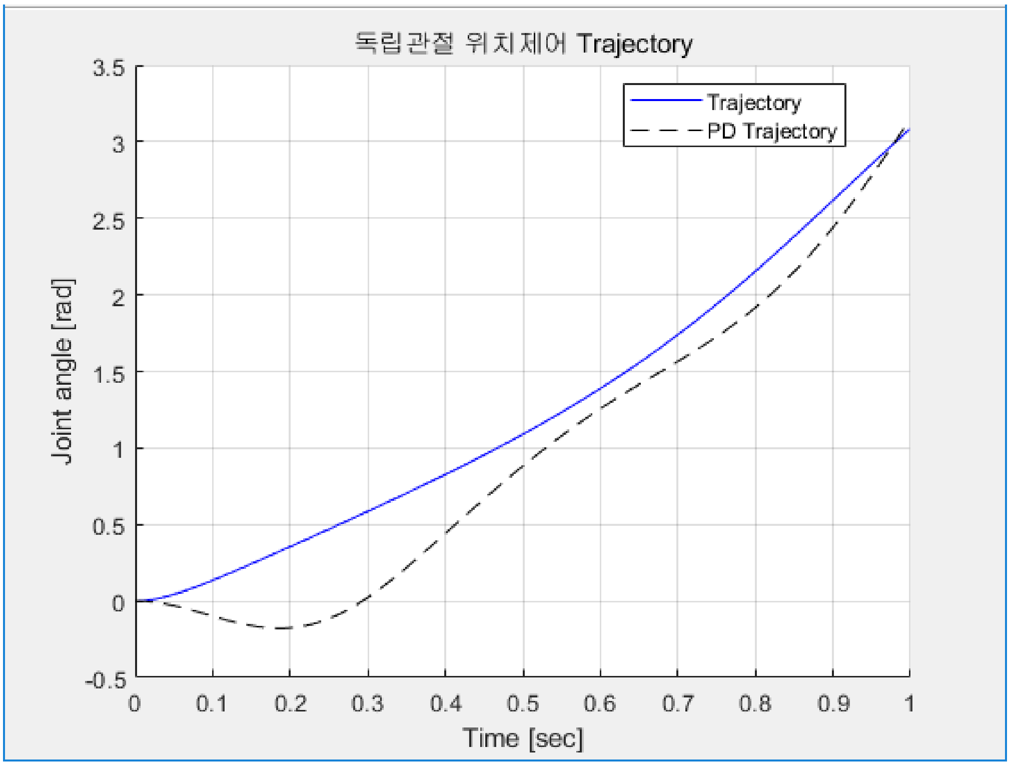


<그림 37> Webot 모델링

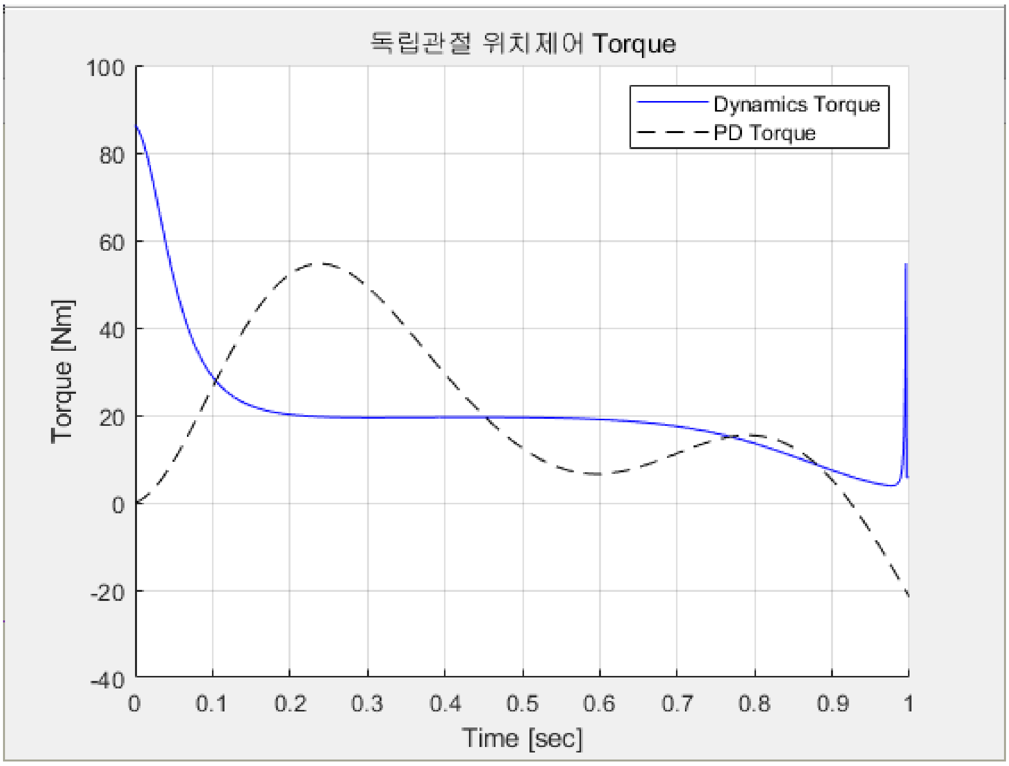
독립관절 위치제어를 하기위해 **Webot**으로 다음과 같이 모델링을 하였다.

모델링을 진행한 이후, PD Control을 하기 위해 P 게인과, D 게인을 넣으며 적절한 게인을 선정하였다.

P 게인과 D 게인이 낮을 때(Kp = 100, Kd = 2), 다음과 같은 경로와 Torque를 보였다.



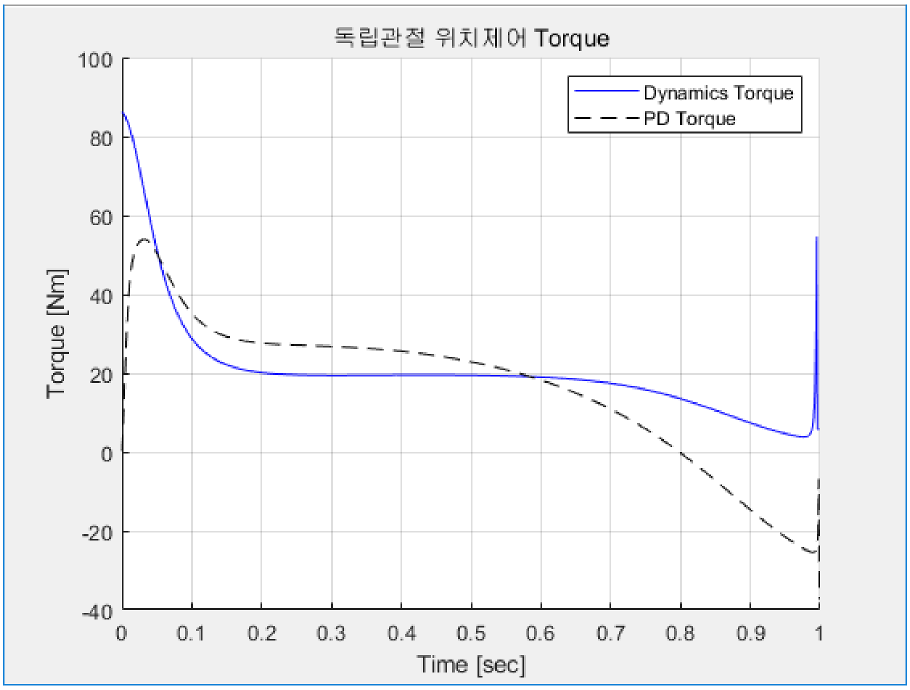
<그림 38> 게인이 낮을 때 Trajectory



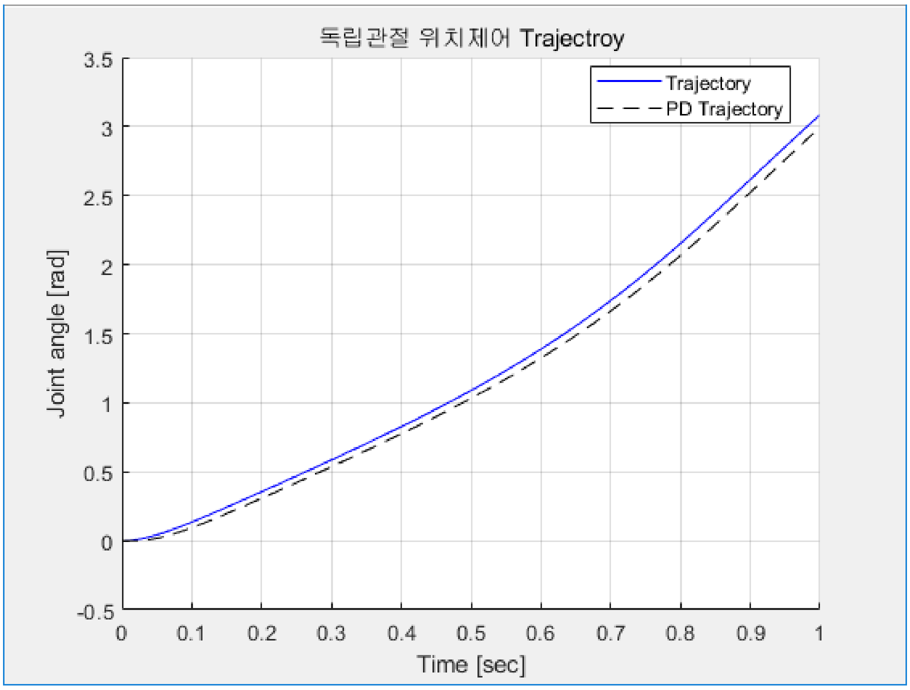
<그림 39> 게인이 낮을 때 Torque

게인이 낮기 때문에, **경로를 잘 추종하지 못하는 것**을 확인하였고, Torque 또한 직접 손으로 계산한 것과 차이가 나는 것을 확인하였다.

적절한 게인을 찾아 게인을 올렸을 때(Kp = 10000, Kd = 200), 다음과 같이 **경로를 잘 추종하는 것**을 볼 수 있었고, Torque값도 손으로 계산한 것과 비슷하게 나오는 것을 볼 수



<그림 40> 게인이 높을 때 Trajectory

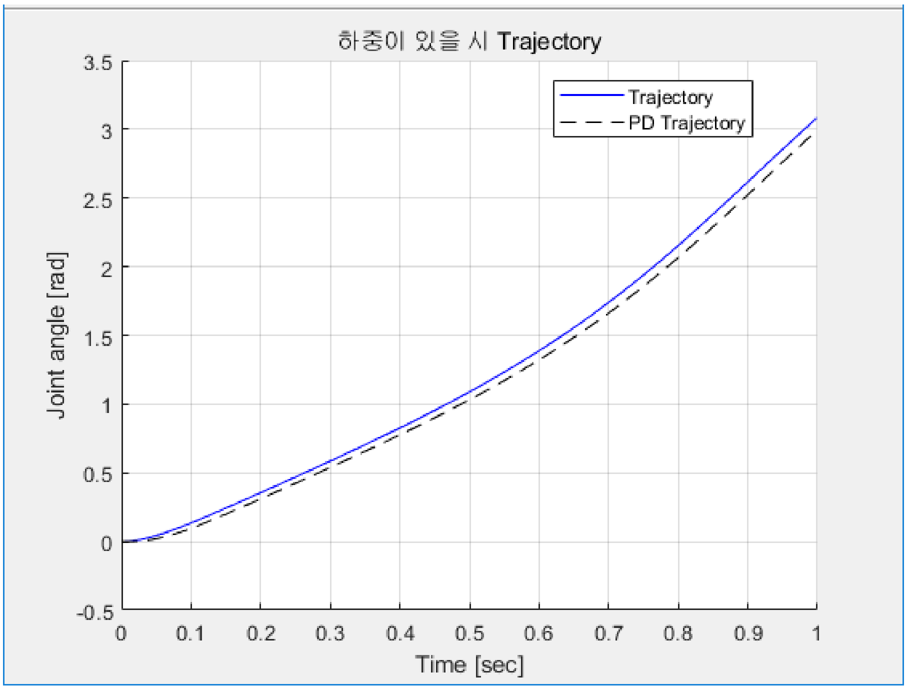


<그림 41> 게인이 높을 때 Torque

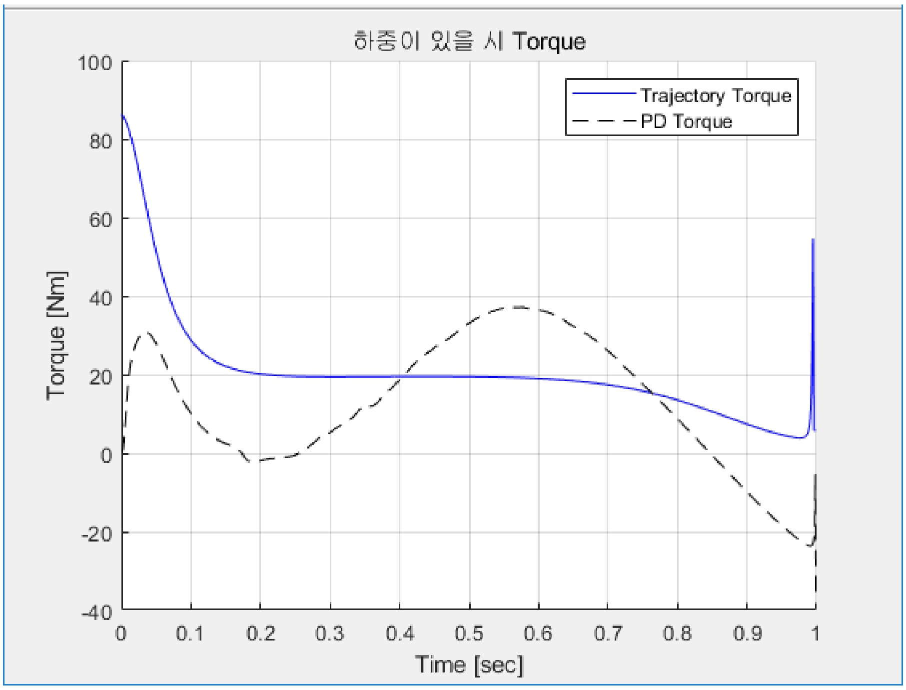
있었다.

그런데 Torque 값이 조금 다른 현상을 보였는데, 직접 계산한 값은 약 88N에서 시작하였지만, 모델링에서는 Torque가 0으로 시작했기 때문에 초반부에 차이가 생긴 것으로 판단되고, 0.6초 이후에 Torque 값이 작아지면서 마이너스 값으로 떨어지게 되는데, 이는 모델링에서 Singularity에서 멈춰야 하기 때문에 반대로 Torque가 작용한 것으로 판단이 된다. 따라서 이러한 문제는 로봇이 중간에 멈추는 것이 아닌 계속해서 운동할 때 없어질 것으로 판단된다.

다음으로 PD 제어로 로봇이 운동할 때 발판부에 일정한 하중이 걸린 것처럼 실험을 하였는데, 다음과 같이 경로 추종은 잘 되었으나, Torque값이 변하는 것을 볼 수 있었다.



<그림 42> 하중이 있을 때 Tajectory

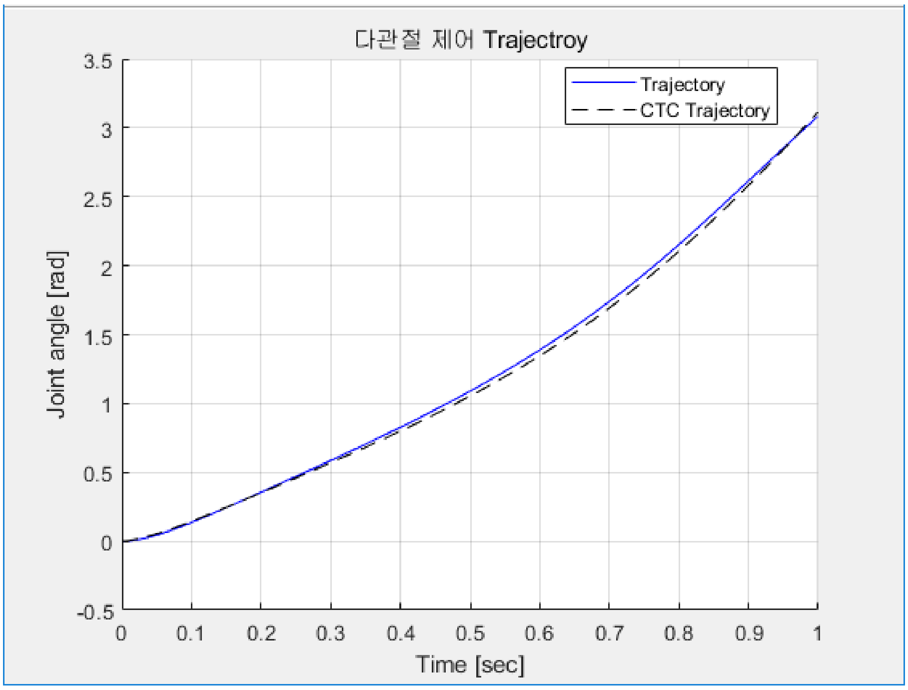


<그림 43> 하중이 있을 때 Torque

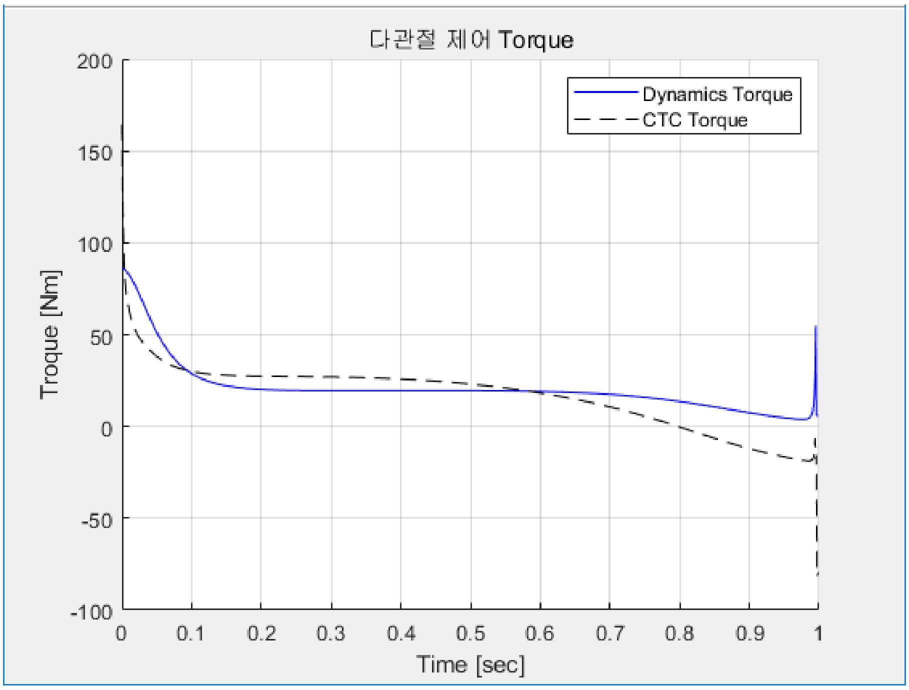
10. 다관절 위치제어

다관절 위치제어를 하기위해 PD 제어와 동일하게 Webot을 이용하여 진행하였으며, **Joint Space**에서의 제어를 기반으로 Torque를 계산하였다. 각 게인을 적절하게 선정하고 제어를 하였을 때 다음과 같이 경로와 Torque를 구할 수 있었다.

이 때 게인은 (Kp = 150, Kv = 70) 으로 설정하였다.

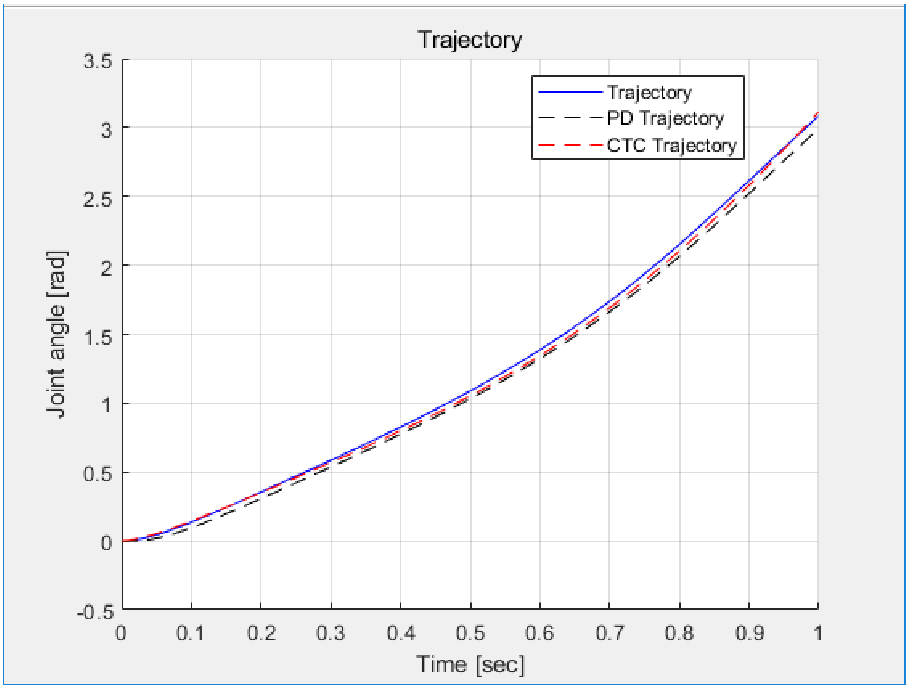


<그림 44> 다관절 위치제어 Trajectory

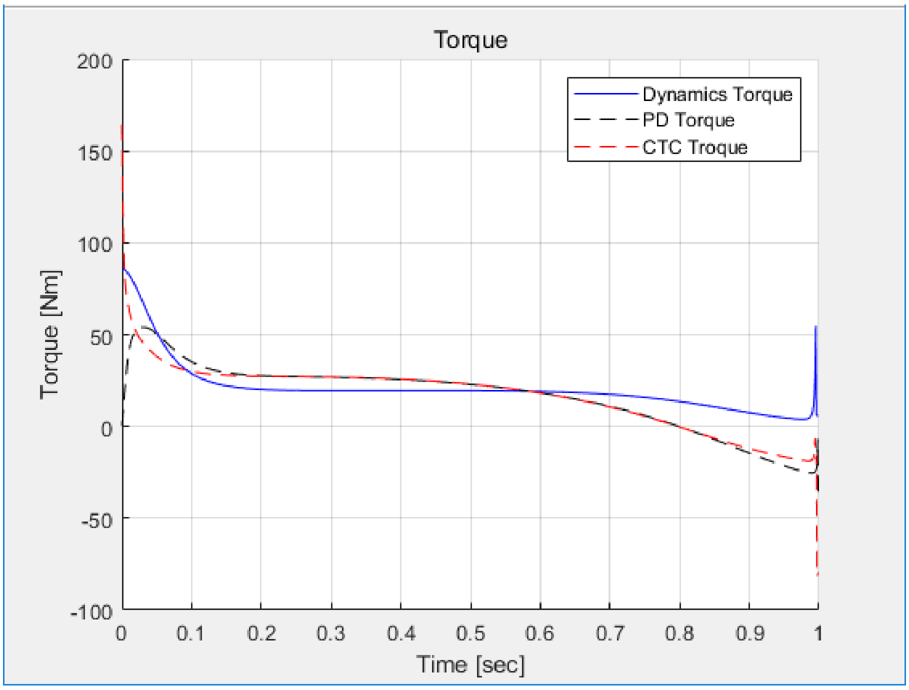


<그림 45> 다관절 위치제어 Torque

이를 PD Control과 비교하기 위해 같이 도식하면 다음과 같다.



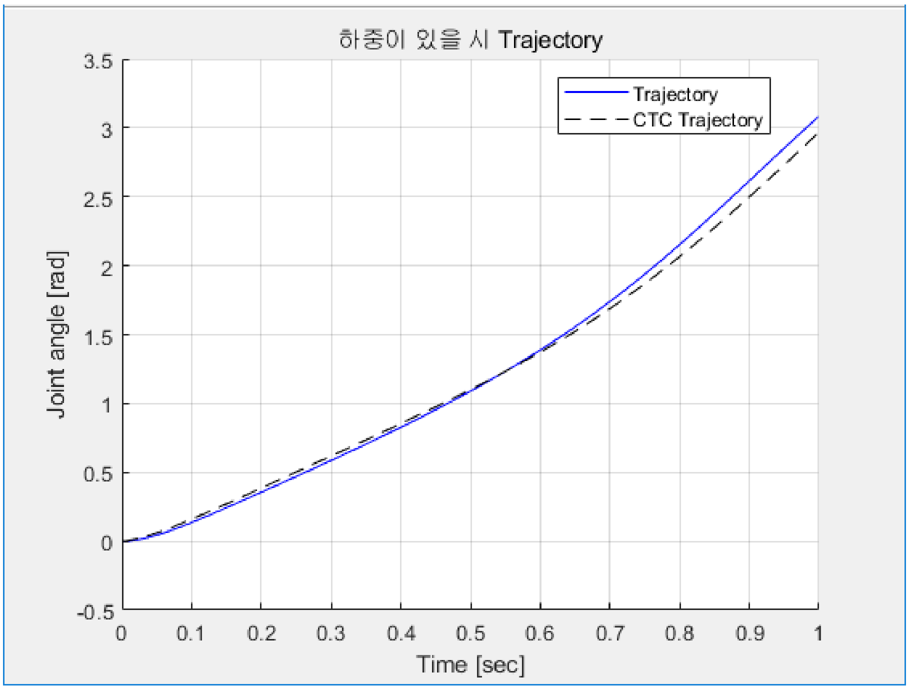
<그림 46> PD 제어 / CTC 제어 Trajectory



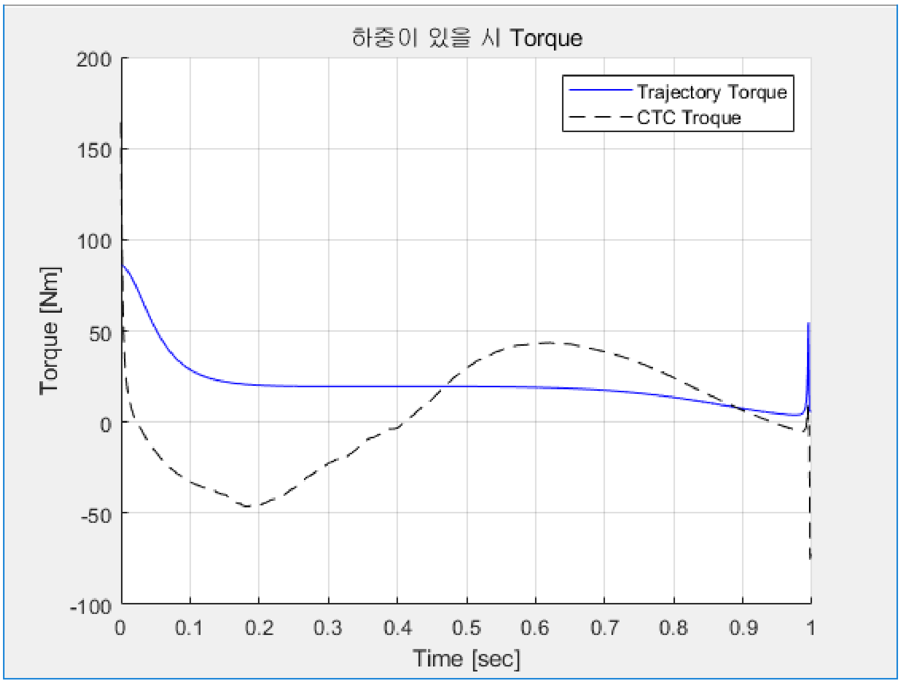
<그림 47> PD 제어 / CTC 제어 Torque

이를 볼 때 **CTC Control이 PD Control보다 더 정확한 Torque 값**이 나오는 것을 볼 수 있었다.

다관절 독립제어에서도 마찬가지로 발판에 사람에 의한 하중이 작용했을 때 Trajectory와 Torque를 구하였다.



<그림 48> 하중에 있을 때 CTC 제어 Trajectory



<그림 49> 하중에 있을 때 CTC 제어 Torque

여기서 PD 제어와는 다르게 경로를 따라가기는 하나 **조금의 오차**가 생기는 것을 볼 수

있고, Torque 값은 PD 제어와 비슷한 경향을 보이고 있다.

마지막으로 PD Control과 CTC Control에서 일정한 힘을 가했을 때 로봇의 **Stiffness**를 확인하였는데, PD Control은 하중에 대해 굉장히 강력한 편이었으나, CTC Control은 하중에 대해 PD Control보다 조금 더 유연한 모습을 보였다.

11. 결론

1. Serial Robot 이 아닌 **Parallel Robot**에서의 동역학 해석에 대해서 공부할 수 있었다.

2. 추후 연구 시에는 경로계획에서 로봇의 움직임이 **사람의 보행 Trajectory**와 유사하도록 Trajectory 를 만들어야 한다.

3. 동역학 해석 시 링크를 단순화하여 질량 관성 모멘트와 질량 중심을 구하였는데, **실제 링크의 질량 중심과 관성 모멘트**를 고려해야 한다.

4. 직접 계산한 Torque와 독립관절제어와 다관절제어에서의 Torque 결과가 상이하게 나왔는데, 크랭크 축이 계속 회전을 하면 이런 문제가 없을 것으로 판단된다. 따라서 **Upper arm과 Under arm**을 계속해서 운동하는 것을 시뮬레이션 하여 Torque값을 구해야 할 것이다.

5. End Effector에 하중이 걸렸을 때 필요한 **모터 토크에 대한 해석**을 더 진행하여, 모터에 필요한 Torque가 어떻게 나오는지 정확한 해석이 필요하다.

12. 부록

A. 매트랩

1. Kinematics.m

```
% %% F.K.
%
% L1 = 185;
% L2 = 438;
% L3 = 176;
% L4 = 454;
%
% th1 = 0;
% th2 = 0;
% th2star = th2 - 15.9233;
% th3 = -54;
%
% A1 = [cosd(th1), -sind(th1), 0, L1*cosd(th1)
%     sind(th1), cosd(th1), 0, L1*sind(th1)
%     0, 0, 1, 0
%     0, 0, 0, 1];
%
% A2 = [cosd(th2star), -sind(th2star), 0, L2*cosd(th2star)
%     sind(th2star), cosd(th2star), 0, L2*sind(th2star)
%     0, 0, 1, 0
%     0, 0, 0, 1];
%
% A3 = [cosd(th3), -sind(th3), 0, L3*cosd(th3)
%     sind(th3), cosd(th3), 0, L3*sind(th3)
%     0, 0, 1, 0
%     0, 0, 0, 1];
%
% T2 = A1*A2
% T3 = A1*A2*A3;
%
% %% I.K.
% xc = 606.1939;
% yc = -120.1654;
% beta = atan2(yc,xc);
%
% D1 = (xc^2 + yc^2 + L1^2 - L2^2) / (2 * sqrt(xc^2 + yc^2) * L1);
% alpha = atan2(sqrt(1-D1^2), D1);
%
```

```
% I_K_th1 = alpha + beta;
% I_K_th1 = I_K_th1 * 180 / pi
%
% D2 = (L1^2 + L2^2 - xc^2 - yc^2) / (2 * L1 * L2);
% gamma = atan2(sqrt(1-D2^2), D2);
% gamma = gamma * 180 / pi;
%
% I_K_th2 = gamma - 180 + 15.9233

%% Animation
clc, clear;

k = 0;
a = 0;
b = 0;
c = 0;

L1 = 185;
L2 = 438;
L3 = 176;
L4 = 454;

th3 = -54;
% upper arm
for i=-7.4302:0.01:40.5160

    k = k + 1;
    a = a + 1;
    xc(k) = 547.5 - L4 * sind(i);
    yc(k) = 330 - L4 * cosd(i);

    beta(k) = atan2(yc(k),xc(k));

    D1 = (xc(k)^2 + yc(k)^2 + L1^2 - L2^2) / (2 * sqrt(xc(k)^2 + yc(k)^2) *
L1);
    alpha(k) = atan2(sqrt(1-D1^2), D1);

    I_K_th1(k) = alpha(k) + beta(k);
    I_K_th1(k) = I_K_th1(k) * 180 / pi;

    D2 = (L1^2 + L2^2 - xc(k)^2 - yc(k)^2) / (2 * L1 * L2);
    gamma(k) = atan2(sqrt(1-D2^2), D2);
    gamma(k) = gamma(k) * 180 / pi;

    I_K_th2_star(k) = gamma(k) - 180;
    I_K_th2(k) = I_K_th2_star(k) + 15.9233;


    if(I_K_th1(k) < -180)
        I_K_th1(k) = 360 + I_K_th1(k);
    end
    if(I_K_th1(k) > 180)
        I_K_th1(k) = -360 + I_K_th1(k);
    end
```

```
    if(I_K_th2(k) < -180)
        I_K_th2(k) = 360 + I_K_th2(k);
    end
    if(I_K_th2(k) > 180)
        I_K_th2(k) = -360 + I_K_th2(k);
    end

    A1 = [cosd(I_K_th1(k)), -sind(I_K_th1(k)), 0, L1*cosd(I_K_th1(k))
        sind(I_K_th1(k)), cosd(I_K_th1(k)), 0, L1*sind(I_K_th1(k))
        0, 0, 1, 0
        0, 0, 0, 1];

    A2 = [cosd(I_K_th2_star(k)), -sind(I_K_th2_star(k)), 0,
L2*cosd(I_K_th2_star(k))
        sind(I_K_th2_star(k)), cosd(I_K_th2_star(k)), 0,
L2*sind(I_K_th2_star(k))
        0, 0, 1, 0
        0, 0, 0, 1];

    A3 = [cosd(th3), -sind(th3), 0, L3*cosd(th3)
        sind(th3), cosd(th3), 0, L3*sind(th3)
        0, 0, 1, 0
        0, 0, 0, 1];

    T2 = A1*A2;
    T3 = A1*A2*A3;

    X1(k) = A1(1, 4);
    Y1(k) = A1(2, 4);
    X2(k) = T2(1, 4);
    Y2(k) = T2(2, 4);
    X3(k) = T3(1, 4);
    Y3(k) = T3(2, 4);
end


% under arm
for i=40.5160:-0.01:-8.0933

    k = k + 1;
    b = b + 1;
    xc(k) = 547.5 - L4 * sind(i);
    yc(k) = 330 - L4 * cosd(i);

    beta(k) = atan2(yc(k),xc(k));

    D1 = (xc(k)^2 + yc(k)^2 + L1^2 - L2^2) / (2 * sqrt(xc(k)^2 + yc(k)^2) *
L1);
    alpha(k) = atan2(-sqrt(1-D1^2), D1);

    I_K_th1(k) = alpha(k) + beta(k);
    I_K_th1(k) = I_K_th1(k) * 180 / pi;

    D2 = (L1^2 + L2^2 - xc(k)^2 - yc(k)^2) / (2 * L1 * L2);
    gamma(k) = atan2(-sqrt(1-D2^2), D2);
```

```
    gamma(k) = gamma(k) * 180 / pi;

    I_K_th2_star(k) = gamma(k) + 180;
    I_K_th2(k) = I_K_th2_star(k) + 15.9233;


    if(I_K_th1(k) < -180)
       I_K_th1(k) = 360 + I_K_th1(k);
    end
    if(I_K_th1(k) > 180)
       I_K_th1(k) = -360 + I_K_th1(k);
    end

    if(I_K_th2(k) < -180)
       I_K_th2(k) = 360 + I_K_th2(k);
    end
    if(I_K_th2(k) > 180)
       I_K_th2(k) = -360 + I_K_th2(k);
    end

    A1 = [cosd(I_K_th1(k)), -sind(I_K_th1(k)), 0, L1*cosd(I_K_th1(k))
       sind(I_K_th1(k)), cosd(I_K_th1(k)), 0, L1*sind(I_K_th1(k))
       0, 0, 1, 0
       0, 0, 0, 1];

    A2 = [cosd(I_K_th2_star(k)), -sind(I_K_th2_star(k)), 0,
L2*cosd(I_K_th2_star(k))
       sind(I_K_th2_star(k)), cosd(I_K_th2_star(k)), 0,
L2*sind(I_K_th2_star(k))
       0, 0, 1, 0
       0, 0, 0, 1];

    A3 = [cosd(th3), -sind(th3), 0, L3*cosd(th3)
       sind(th3), cosd(th3), 0, L3*sind(th3)
       0, 0, 1, 0
       0, 0, 0, 1];

    T2 = A1*A2;
    T3 = A1*A2*A3;

    X1(k) = A1(1, 4);
    Y1(k) = A1(2, 4);
    X2(k) = T2(1, 4);
    Y2(k) = T2(2, 4);
    X3(k) = T3(1, 4);
    Y3(k) = T3(2, 4);
end

% upper arm
for i=-8.0933:0.01:-7.4302

    k = k + 1;
    c = c + 1;
    xc(k) = 547.5 - L4 * sind(i);
    yc(k) = 330 - L4 * cosd(i);
```

```
    beta(k) = atan2(yc(k),xc(k));

    D1 = (xc(k)^2 + yc(k)^2 + L1^2 - L2^2) / (2 * sqrt(xc(k)^2 + yc(k)^2) *
L1);
    alpha(k) = atan2(sqrt(1-D1^2), D1);

    I_K_th1(k) = alpha(k) + beta(k);
    I_K_th1(k) = I_K_th1(k) * 180 / pi;

    D2 = (L1^2 + L2^2 - xc(k)^2 - yc(k)^2) / (2 * L1 * L2);
    gamma(k) = atan2(sqrt(1-D2^2), D2);
    gamma(k) = gamma(k) * 180 / pi;

    I_K_th2_star(k) = gamma(k) - 180;
    I_K_th2(k) = I_K_th2_star(k) + 15.9233;


    if(I_K_th1(k) < -180)
        I_K_th1(k) = 360 + I_K_th1(k);
    end
    if(I_K_th1(k) > 180)
        I_K_th1(k) = -360 + I_K_th1(k);
    end

    if(I_K_th2(k) < -180)
        I_K_th2(k) = 360 + I_K_th2(k);
    end
    if(I_K_th2(k) > 180)
        I_K_th2(k) = -360 + I_K_th2(k);
    end

    A1 = [cosd(I_K_th1(k)), -sind(I_K_th1(k)), 0, L1*cosd(I_K_th1(k))
        sind(I_K_th1(k)), cosd(I_K_th1(k)), 0, L1*sind(I_K_th1(k))
        0, 0, 1, 0
        0, 0, 0, 1];

    A2 = [cosd(I_K_th2_star(k)), -sind(I_K_th2_star(k)), 0,
L2*cosd(I_K_th2_star(k))
        sind(I_K_th2_star(k)), cosd(I_K_th2_star(k)), 0,
L2*sind(I_K_th2_star(k))
        0, 0, 1, 0
        0, 0, 0, 1];

    A3 = [cosd(th3), -sind(th3), 0, L3*cosd(th3)
        sind(th3), cosd(th3), 0, L3*sind(th3)
        0, 0, 1, 0
        0, 0, 0, 1];

    T2 = A1*A2;
    T3 = A1*A2*A3;

    X1(k) = A1(1, 4);
    Y1(k) = A1(2, 4);
    X2(k) = T2(1, 4);
```

```
    Y2(k) = T2(2, 4);
    X3(k) = T3(1, 4);
    Y3(k) = T3(2, 4);
end


% axis([-200, 700, -350, 100])
axis([200, 700, -200, 100])
axis equal
hold on

% plot(0,0, 'o')
% plot(X1, Y1)
plot(X2, Y2)
% plot(X3, Y3)

% for j = 1:974
%
%     line1 = line([0, X1(j)], [0, Y1(j)]);
%     line2 = line([X1(j), X2(j)], [Y1(j), Y2(j)]);
%     line3 = line([X2(j), X3(j)], [Y2(j), Y3(j)]);
%     line4 = line([X2(j), 547.5], [Y2(j), 330])
%
%     pause(0.01)
%     delete(line1)
%     delete(line2)
%     delete(line3)
%     delete(line4)
%
% end

% time = 1:k;
% plot(time,X2)
% hold on
% plot(time,Y2)



% plot(X2, Y2)
%
% for j = 1:1:480
%
%     P = plot(X2(j),Y2(j), 'bo');
%     pause(0.01)
%     delete(P)
%
% end
% plot(X2(j), Y2(j), 'b*')
% for j = 481:1:967
%
%     P = plot(X2(j),Y2(j), 'ko');
%     pause(0.01)
%     delete(P)
%
% end
% plot(X2(j), Y2(j), 'k*')
```

```
% for j = 968:1:974
%
%     P = plot(X2(j),Y2(j), 'ro');
%     pause(0.01)
%     delete(P)
%
% end
%
% plot(X2(j), Y2(j), 'r*');

y0_dot = (Y2(1)-Y2(2)) / (X2(1)-X2(2))
yf_dot = (Y2(4795)-Y2(4794)) / (X2(4795)-X2(4794))

for j = 1:4795
    XJ(j) = X2(j);
    YJ(j) = Y2(j);
end

plot(XJ, YJ)
```

2. Kinematics_th.m

```
clc, clear;

L1 = 185;
L2 = 438;
L3 = 176;
L4 = 454;

th3 = -54;

k = 0;


for i = 0:0.1:180

    k = k+1;
    th1(k) = i;
    L = sqrt((547.5 - L1*cosd(th1(k)))^2 + (330 - L1*sind(th1(k)))^2);

    D1 = (L2^2 + L^2 - L4^2) / (2 * L2 * L);

    alpha = atan2( sqrt(1-D1^2), D1);
    alpha = alpha * 180 / pi;

    beta = atan2(547.5 - L1 * cosd(th1(k)), 330 - L1 * sind(th1(k)));
    beta = beta * 180 / pi;

    th2_star(k) = (90 - th1(k)) - (alpha + beta);

    if(th1(k) < -180)
        th1(k) = 360 + th1(k);
    end
    if(th1(k) > 180)
```

```
        th1(k) = -360 + th1(k);
    end

    if(th2_star(k) < -180)
        th2_star(k) = 360 + th2_star(k);
    end
    if(th2_star(k) > 180)
        th2_star(k) = -360 + th2_star(k);
    end

    D2 = (L2^2 + L4^2 - L^2) / (2*L2*L4);
    th4_star(k) = atan2(sqrt(1-D2^2), D2);
    th4_star(k) = 180 - th4_star(k) * 180 / pi;

    A1 = [cosd(th1(k)), -sind(th1(k)), 0, L1*cosd(th1(k))
        sind(th1(k)), cosd(th1(k)), 0, L1*sind(th1(k))
        0, 0, 1, 0
        0, 0, 0, 1];

    A2 = [cosd(th2_star(k)), -sind(th2_star(k)), 0, L2*cosd(th2_star(k))
        sind(th2_star(k)), cosd(th2_star(k)), 0, L2*sind(th2_star(k))
        0, 0, 1, 0
        0, 0, 0, 1];

    A3 = [cosd(th3), -sind(th3), 0, L3*cosd(th3)
        sind(th3), cosd(th3), 0, L3*sind(th3)
        0, 0, 1, 0
        0, 0, 0, 1];

    A4 = [cosd(th4_star(k)), -sind(th4_star(k)), 0, L4*cosd(th4_star(k))
        sind(th4_star(k)), cosd(th4_star(k)), 0, L4*sind(th4_star(k))
        0, 0, 1, 0
        0, 0, 0, 1];

    T2 = A1*A2;
    T3 = A1*A2*A3;
    T4 = A1*A2*A4;

    X1(k) = A1(1, 4);
    Y1(k) = A1(2, 4);
    X2(k) = T2(1, 4);
    Y2(k) = T2(2, 4);
    X3(k) = T3(1, 4);
    Y3(k) = T3(2, 4);
    X4(k) = T4(1, 4);
    Y4(k) = T4(2, 4);

end

% axis([-200, 700, -200, 100])
% axis equal
% hold on
%
% plot(0,0, 'o')
% plot(X1, Y1)
```

```
% plot(X2, Y2)
% plot(X3, Y3)
%
% for j = 1:1801
%
%    line1 = line([0, X1(j)], [0, Y1(j)]);
%    line2 = line([X1(j), X2(j)], [Y1(j), Y2(j)]);
%    line3 = line([X2(j), X3(j)], [Y2(j), Y3(j)]);
%    line4 = line([X2(j), 547.5], [Y2(j), 330]);
%
%    pause(0.01)
%    delete(line1)
%    delete(line2)
%    delete(line3)
%    delete(line4)
%
% end
```

3. Trajectory.m

```
clc, clear;


L1 = 185;
L2 = 438;
L3 = 176;
L4 = 454;

th3 = -57*pi/180;

% 1st trajectory

% t0 = 0;
% tf = 0.7;
% q0 = -120.1654;
% v0 = 0;
% qf = -15.1531;
% vf = 0;
%
% A = [1, t0, t0^2, t0^3;
%    0, 1, 2*t0, 3*t0^2;
%    1, tf, tf^2, tf^3;
%    0, 1, 2*tf, 3*tf^2];
%
% B = inv(A) * [q0, v0, qf, vf]';

i = 0;

for t = 0:0.001:1
    i = i+1;
    T(i) = t;
    X(i) = (606.1939 - 1060.9443*t^2 + 707.2962*t^3);
    Y(i) = 330 - sqrt(206116 - (58.6939 - 1060.94*t^2 + 707.296*t^3)^2);

    X_dot(i) = (-2121.89 * t + 2121.89 * t^2);
```

```
    Y_dot(i) = (-2121.89*t + 2121.89*t^2) * (58.6939 - 1060.94 * t^2 +
707.296 * t^3) / sqrt(206116 - (58.6939 - 1060.94 * t^2 + 707.296 *
t^3)^2);

    X_dotdot(i) = -4330.89 + 4243.78*t;
    Y_dotdot(i) = ( (-2121.89*t + 2121.89*t^2)^2 * (58.6939 - 1060.94*t^2 +
707.296*t^3)^2 / (206116 - (58.6939 - 1060.94 * t^2 + 707.296 *
t^3)^2)^(3/2) ) + ( (-2121.89*t + 2121.89*t^2)^2 / sqrt(206116 - (58.6939 -
1060.94*t^2 + 707.296 * t^3)^2) ) + ( (-2121.89 + 4243.78*t) * (58.6939 -
1060.94 * t^2 + 707.296 * t^3) / sqrt(206116 - (58.6939 - 1060.94 * t^2 +
707.296*t^3)^2) );
end


% for t = 0:0.01:0.7
%     i = i+1;
%     T(i) = t;
%     X(i) = (606.1939 - 2165.1924*t^2 + 2062.0880*t^3);
%     Y(i) = 330 - sqrt(206116 - (58.6939 - 2165.19*t^2 + 2062.09*t^3)^2);
%
%     X_dot(i) = (-2 * 2165.1924 * t + 3 * 2062.0880 * t^2);
%     Y_dot(i) = (-4330.38*t + 6186.26*t^2) * (58.6939 - 2165.19 * t^2 +
2062.09 * t^3) / sqrt(206116 - (58.6939 - 2165.19*t^2 + 2062.09*t^3)^2);
%
%     X_dotdot(i) = -4330.38 + 12372.5*t;
%     Y_dotdot(i) = ( (-4330.38*t + 6186.26*t^2)^2 * (58.6939 - 2165.19*t^2
+ 2062.09*t^3)^2 / (206116 - (58.6939 - 2165.19 * t^2 + 2062.09 *
t^3)^2)^(3/2) ) + ( (-4330.38*t + 6186.26*t^2)^2 / sqrt(206116 - (58.6939 -
2165.19*t^2 + 2062.09 * t^3)^2) ) + ( (-4330.38 + 12372.5*t) * (58.6939 -
2165.19 * t^2 + 2062.09 * t^3) / sqrt(206116 - (58.6939 - 2165.19 * t^2 +
2062.09*t^3)^2) );
% end
%
% % 2nd trajectory
%
% t0_2 = 0.7;
% tf_2 = 1.6;
% q0_2 = 252.5458;
% v0_2 = 0;
% qf_2 = 611.4433;
% vf_2 = 0;
%
% % i = 0;
% for t = 0.7:0.01:1.6
%     i = i+1;
%     T(i) = t;
%     X(i) = (252.5458 + 1329.25*(t - t0_2)^2 - 984.6296*(t - t0_2)^3);
%     Y(i) = 330 - sqrt(454^2 - (X(i)-547.5)^2);
%
%     X_dot(i) = (2*1329.25*(t-t0_2) - 3*984.6296*(t-t0_2)^2);
%     Y_dot(i) = ( (2658.5*(t-t0_2) - 2953.89 * (t-t0_2)^2) * (-294.954 +
1329.25 * (t-t0_2)^2 - 984.63 * (t-t0_2)^3) ) / ( sqrt(206116 - (-294.954 +
1329.25 * (t-t0_2)^2 - 984.63 * (t-t0_2)^3)^2) );
%
%     X_dotdot(i) = 2658.5 - 5907.78 * (t-t0_2);
%     Y_dotdot(i) = ( (2658.5*(t-t0_2) - 2953.89 * (t-t0_2)^2)^2 /
sqrt(206116 - (-294.954 + 1329.25 * (t-t0_2) ^ 2 -984.63 * (t-t0_2)^3)^2) )
```

```
+ ( (2658.5 - 5907.78*(t-t0_2)) * (-294.954 + 1329.25*(t-t0_2)^2 - 984.63
*(t-t0_2)^3) / sqrt(206116 - (-294.954 + 1329.25 * (t-t0_2)^2 - 984.63 *
(t-t0_2)^3)^2 ) ) + ( (2658.5*(t-t0_2) - 2953.89 * (t-t0_2)^2)^2 * (-
294.954 + 1329.25 *(t-t0_2)^2 - 984.63*(t-t0_2)^3)^2 / (206116 - (-294.954
+ 1329.25 * (t-t0_2)^2 - 984.63 * (t-t0_2)^3)^2)^(3/2) );
%
% end
%
% % plot(T, X2)
%
% % 3nd trajectory
%
% t0_3 = 1.6;
% tf_3 = 1.8;
% q0_3 = 611.4433;
% v0_3 = 0;
% qf_3 = 606.1939;
% vf_3 = 0;
%
% % i = 0;
% for t = 1.6:0.01:1.8
%     i = i+1;
%     T(i) = t;
%     X(i) = (611.4433 - 393.705*(t-t0_3)^2 + 1312.35*(t-t0_3)^3);
%     Y(i) = 330 - sqrt(454^2 - (X(i)-547.5)^2);
%
%     X_dot(i) = (-2*393.705*(t-t0_3) + 3*1312.35*(t-t0_3)^2);
%     Y_dot(i) = ( (-787.41 * (t-t0_3) + 3936.75 * (t-t0_3)^2) * (63.9433 -
393.705 * (t-t0_3)^2 + 1312.25 * (t-t0_3)^3) / sqrt(206116 - (63.9433 -
393.705 *(t-t0_3)^2 + 1312.25 * (t-t0_3)^3)^2) );
%
%     X_dotdot(i) = -787.41+7873.5*(t-t0_3);
%     Y_dotdot(i) = ( (-787.41 * (t-t0_3) + 3936.75 * (t-t0_3)^2)^2 /
sqrt(206116 - (63.9433 - 393.705 * (t-t0_3)^2 + 1312.25 * (t-t0_3)^3)^2) )
+ ( (-787.41 + 7873.5 * (t-t0_3)) * (63.9433 - 393.705 * (t-t0_3)^2 +
1312.25 * (t-t0_3)^3) / sqrt(206116 - (63.9433 - 393.705 * (t-t0_3)^2 +
1312.25 * (t-t0_3)^3)^2) ) + ( (-787.41 * (t-t0_3) + 3936.75 * (t-
t0_3)^2)^2 * (63.9433 - 393.705 * (t-t0_3)^2 + 1312.25 * (t-t0_3)^3)^2 /
(206116 - ( 63.9433 -393.705*(t-t0_3)^2 + 1312.25 * (t-t0_3)^3)^2)^(3/2) );
%
% end
% ylabel("Y acceleration [mm/s^2]")
% xlabel("Time [sec]")
% title('E-E Y acceleration');
% grid on;
% hold on;
% legend
% plot(T, X)
% figure
% plot(T, X_dot)
% figure
% plot(T, X_dotdot)
% figure
% plot(T, Y)
% figure
% plot(T, Y_dot)
% figure
```

```
% plot(T, Y_dotdot)


%% I.K. ·Î th1, th2 ±¸ÇÏ±â
% upper arm

for k = 1:1001
    beta(k) = atan2(Y(k),X(k));

    D1(k) = (X(k)^2 + Y(k)^2 + L1^2 - L2^2) / (2 * sqrt(X(k)^2 + Y(k)^2) *
L1);

    if(D1(k) >= 1)
        D1(k) = 0.99999999999;
    end
    if(D1(k) <= -1)
        D1(k) = -0.9999999999;
    end

    alpha(k) = atan2(sqrt(1-D1(k)^2), D1(k));

    I_K_th1(k) = alpha(k) + beta(k);
    I_K_th1(k) = I_K_th1(k);

    D2(k) = (L1^2 + L2^2 - X(k)^2 - Y(k)^2) / (2 * L1 * L2);

    if(D2(k) >= 1)
        D2(k) = 0.99999999999;
    end
    if(D2(k) <= -1)
        D2(k) = -0.9999999999;
    end

    gamma(k) = atan2(sqrt(1-D2(k)^2), D2(k));
    gamma(k) = gamma(k);

    I_K_th2_star(k) = gamma(k) - pi;
    I_K_th2(k) = I_K_th2_star(k) + (15.9233*pi/180);

    L(k) = sqrt((547.5 - 185*cos(I_K_th1(k)))^2 + (330 -
185*sin(I_K_th1(k)))^2);

    D3(k) = (L2^2 + L4^2 - L(k)^2) / (2*L2*L4);
    I_K_th4_star(k) = atan2(sqrt(1-D3(k)^2), D3(k));
    I_K_th4_star(k) = 2*pi - I_K_th4_star(k);
    I_K_th4(k) = I_K_th4_star(k) - (113.3539*pi/180);

    if(I_K_th1(k) < -pi)
        I_K_th1(k) = 2*pi + I_K_th1(k);
    end
    if(I_K_th1(k) > pi)
        I_K_th1(k) = -2*pi + I_K_th1(k);
    end
```

```
    if(I_K_th2(k) < -pi)
        I_K_th2(k) = 2*pi + I_K_th2(k);
    end
    if(I_K_th2(k) > pi)
        I_K_th2(k) = -2*pi + I_K_th2(k);
    end

    if(I_K_th4(k) < -pi)
        I_K_th4(k) = 2*pi + I_K_th4(k);
    end
    if(I_K_th4(k) > pi)
        I_K_th4(k) = -2*pi + I_K_th4(k);
    end
end

fname = 'C:\Users\whgdm\OneDrive\¹ÙÅÁ È¸é\¼ö¾÷\1.
·Î°¿µ¿¿ªÇÐ¹×Á¦¾î\webot\controllers\4Link\data2.txt';
A = load(fname);
fname1 = 'C:\Users\whgdm\OneDrive\¹ÙÅÁ È¸é\¼ö¾÷\1.
·Î°¿µ¿¿ªÇÐ¹×Á¦¾î\webot\controllers\4Link\data3.txt';
B = load(fname1);

ylabel("Joint angle [rad]")
xlabel("Time [sec]")
title('ÇÏÁßÀÌ ÀÖÀ» ½Ã Trajectory');
grid on;
hold on;
legend
T = 0:0.001:0.999;
I_K_th1 = -A(:,1);
th1 = -A(:,2);
% % CTC_th1 = -B(:,2);
%
plot(T, I_K_th1, 'b')
% % hold on;
plot(T, th1, '--k')
% th1_dot = -A(:,4);
% th1_dotdot = -A(:,5);

% plot(T, th1, '--k')
% hold on
% plot(T, CTC_th1, '--r')

I_K_th1_dot = diff(I_K_th1) ./ 0.001;
I_K_th1_dot(k) = I_K_th1_dot(k-1);
I_K_th1_dotdot = diff(I_K_th1_dot) ./ 0.001;
I_K_th1_dotdot(k) = I_K_th1_dotdot(k-1);

I_K_th1_dot(999:1001) = 0;
I_K_th1_dotdot(997:1001) = 0;

% plot(T, I_K_th1, 'b')
% figure
% plot(T, I_K_th1_dot)
% figure
% plot(T, I_K_th1_dotdot)
```

```
% figure

% I_K_th2_dot = diff(I_K_th2) ./ 0.001;
% I_K_th2_dot(k) = I_K_th2_dot(k-1);
% I_K_th2_dotdot = diff(I_K_th2_dot) ./ 0.001;
% I_K_th2_dotdot(k) = I_K_th2_dotdot(k-1);
%
% I_K_th2_dot(999:1001) = 0;
% I_K_th2_dotdot(997:1001) = 0;
%
% I_K_th4_dot = diff(I_K_th4) ./ 0.001;
% I_K_th4_dot(k) = I_K_th4_dot(k-1);
% I_K_th4_dotdot = diff(I_K_th4_dot) ./ 0.001;
% I_K_th4_dotdot(k) = I_K_th4_dotdot(k-1);

% plot(T, I_K_th1_dot)
% figure
% plot(T, I_K_th2_dot)
% I_K_th1_dotdot(996:999) = 0;
% plot(T, I_K_th1_dot)
% figure
% plot(T, I_K_th1_dotdot)
% figure

for k=1:1000

    A1 = [cos(I_K_th1(k)), -sin(I_K_th1(k)), 0, L1*cos(I_K_th1(k))
        sin(I_K_th1(k)), cos(I_K_th1(k)), 0, L1*sin(I_K_th1(k))
        0, 0, 1, 0
        0, 0, 0, 1];

    R1 = [cos(I_K_th1(k)), -sin(I_K_th1(k)), 0
        sin(I_K_th1(k)), cos(I_K_th1(k)), 0
        0, 0, 1];

    A2 = [cos(I_K_th2_star(k)), -sin(I_K_th2_star(k)), 0,
L2*cos(I_K_th2_star(k))
        sin(I_K_th2_star(k)), cos(I_K_th2_star(k)), 0,
L2*sin(I_K_th2_star(k))
        0, 0, 1, 0
        0, 0, 0, 1];

    R2 = [cos(I_K_th2_star(k)), -sin(I_K_th2_star(k)), 0
        sin(I_K_th2_star(k)), cos(I_K_th2_star(k)), 0
        0, 0, 1];

    A3 = [cos(th3), -sin(th3), 0, L3*cos(th3)
        sin(th3), cos(th3), 0, L3*sin(th3)
        0, 0, 1, 0
        0, 0, 0, 1];

    R3 = [cos(th3), -sin(th3), 0
        sin(th3), cos(th3), 0
        0, 0, 1];
```

```
    A4 = [cos(I_K_th4_star(k)), -sin(I_K_th4_star(k)), 0,
L4*cos(I_K_th4_star(k))
        sin(I_K_th4_star(k)), cos(I_K_th4_star(k)), 0,
L4*sin(I_K_th4_star(k))
        0, 0, 1, 0
        0, 0, 0, 1];

    R4 = [cos(I_K_th4_star(k)), -sin(I_K_th4_star(k)), 0
        sin(I_K_th4_star(k)), cos(I_K_th4_star(k)), 0
        0, 0, 1];

    T2 = A1*A2;
    T3 = A1*A2*A3;
    T4 = A1*A2*A4;

    X1(k) = A1(1, 4);
    Y1(k) = A1(2, 4);
    X2(k) = T2(1, 4);
    Y2(k) = T2(2, 4);
    X3(k) = T3(1, 4);
    Y3(k) = T3(2, 4);
    X4(k) = T4(1, 4);
    Y4(k) = T4(2, 4);

end

% axis([-200, 700, -400, 200])
% axis equal
% hold on
%
% plot(0,0, 'o')
% plot(X1, Y1)
% plot(X2, Y2)
% plot(X3, Y3)
%
% for j = 1:1001
%
%     line1 = line([0, X1(j)], [0, Y1(j)]);
%     line2 = line([X1(j), X2(j)], [Y1(j), Y2(j)]);
%     line3 = line([X2(j), X3(j)], [Y2(j), Y3(j)]);
%     line4 = line([X2(j), 547.5], [Y2(j), 330]);
%
%     pause(0.01)
%     delete(line1)
%     delete(line2)
%     delete(line3)
%     delete(line4)
%
% end


%% Eular

L1 = 0.185;
L2 = 0.438;
L3 = 0.176;
```

```
L4 = 0.454;

x = 0.5475;
y = 0.33;

Ic0zz = 0.1318;
Ic1zz = 0.0275;
Ic2zz = 0.00223;
Ic3zz = 0.04172;

m1 = 1.7;
m2 = 0.8;
m3 = 2.4;
g = 9.81;

for j=1:1000
%     torque_E1(j) = Ic1zz + Ic2zz + L1^2 * (m1+m2) + 0.25 * (L3^2*m2 + L2^2
* (m1+4*m2)) + g*L1*(m1+m2)*cos(I_K_th1(j)) +
L1*L2*(m1+2*m2)*cos(I_K_th2(j)) + 0.5*g*L2*m1*cos(I_K_th1(j)+I_K_th2(j)) +
g*L2*m2*cos(I_K_th1(j)+I_K_th2(j)) + L2*L3*m2*cos(th3) +
L1*L3*m2*cos(I_K_th2(j) + th3) +
0.5*g*L3*m2*cos(I_K_th1(j)+I_K_th2(j)+th3);
%     torque_E2(j) = Ic1zz + Ic2zz + Ic3zz + g*m1*(L1*cos(I_K_th1(j)) +
L2*cos(I_K_th1(j)+I_K_th2(j)/2)) + g*m2*(L1*cos(I_K_th1(j)) +
L2*cos(I_K_th1(j)+I_K_th2(j)) + L3*cos(I_K_th1(j)+I_K_th2(j)+th3)/2) +
g*m3*(L1*cos(I_K_th1(j)) + L2*cos(I_K_th1(j)+I_K_th2(j)) +
L4*sin(I_K_th1(j)+I_K_th2(j)+I_K_th4(j))/2) + m3*(1/4*(x+L1*cos(I_K_th1(j))
+ L2*cos(I_K_th1(j) + I_K_th2(j)))^2 + 0.25*(-y-L1*sin(I_K_th1(j)) -
L2*sin(I_K_th1(j) + I_K_th2(j)))^2) + m1*((L1*cos(I_K_th1(j)) + 0.5 *
L2*cos(I_K_th1(j)+I_K_th2(j)))^2 + (-L1*sin(I_K_th1(j)) - 0.5 * L2 *
sin(I_K_th1(j) + I_K_th2(j)))^2) + 0.25 * m3 * (L1*y*cos(I_K_th1(j)) +
L2*y*cos(I_K_th1(j) + I_K_th2(j)) - x*(L1*sin(I_K_th1(j)) + L2 *
sin(I_K_th1(j) + I_K_th2(j)))) + m2 * ((L1 * cos(I_K_th1(j)) + L2 *
cos(I_K_th1(j) + I_K_th2(j)) + 0.5*L3*cos(I_K_th1(j) + I_K_th2(j) + th3))^2
+ (-L1*sin(I_K_th1(j)) - L2*sin(I_K_th1(j) + I_K_th2(j)) -
0.5*L3*sin(I_K_th1(j) + I_K_th2(j) + th3))^2);
    torque_E3(j) =
0.25*(4*Ic0zz+4*Ic1zz+4*Ic2zz+4*Ic3zz+L3^2*m2+L2^2*(m1+4*m2+m3)+L1^2*(4*(m1
+m2)+m3)+m3*(x^2+y^2)+2*(L1*m3*x*cos(I_K_th1(j))+L1*L2*(2*m1+4*m2+m3)*cos(I
_K_th2(j))+L2*m3*x*cos(I_K_th1(j)+I_K_th2(j))+2*L2*L3*m2*cos(th3)+2*L1*L3*m
2*cos(I_K_th2(j)+th3)+L1*m3*y*sin(I_K_th1(j))+L2*m3*y*sin(I_K_th1(j)+I_K_th
2(j))))*I_K_th1_dotdot(j)+0.25*m3*(L1*y*cos(I_K_th1(j))+L2*y*cos(I_K_th1(j)
+I_K_th2(j))-
x*(L1*sin(I_K_th1(j))+L2*sin(I_K_th1(j)+I_K_th2(j))))*I_K_th1_dot(j)+0.5*g*
(L1*(2*(m1+m2)+m3)*cos(I_K_th1(j))+L2*(m1+2*m2+m3)*cos(I_K_th1(j)+I_K_th2(j
))+L3*m2*cos(I_K_th1(j)+I_K_th2(j)+th3));
%     torque_E3(j) =
0.25*m3*(L1*y*cos(I_K_th1(j))+L2*y*cos(I_K_th1(j)+I_K_th2(j))-
x*(L1*sin(I_K_th1(j))+L2*sin(I_K_th1(j)+I_K_th2(j))))*I_K_th1_dot(j)
%     tourqe_E3(j) =
0.5*g*(L1*(2*(m1+m2)+m3)*cos(I_K_th1(j))+L2*(m1+2*m2+m3)*cos(I_K_th1(j)+I_K
_th2(j))+L3*m2*cos(I_K_th1(j)+I_K_th2(j)+th3));
end
% T = 0:0.001:0.999;
% plot(T,torque_E3, 'b')
% hold on
```

```
% plot(T,torque_E2)
% figure

% webot_torque = [-3.40000000000000e-05,-17.3668680000000,-
19.4600460000000,-23.7752690000000,-27.1473550000000,-29.4915410000000,-
30.9866450000000,-31.8283010000000,-32.1934000000000,-32.2323960000000,-
32.0678770000000,-31.7955570000000,-31.4867840000000,-31.1917940000000,-
30.9432860000000,-30.7599690000000,-30.6498680000000,-30.6132510000000,-
30.6451130000000,-30.7371990000000,-30.8795850000000,-31.0618560000000,-
31.2739350000000,-31.5066290000000,-31.7519350000000,-32.0031790000000,-
32.2550150000000,-32.5033520000000,-32.7452100000000,-32.9785650000000,-
33.2021700000000,-33.4153940000000,-33.6180680000000,-33.8103520000000,-
33.9926260000000,-34.1653990000000,-34.3292420000000,-34.4847380000000,-
34.6324470000000,-34.7728820000000,-34.9065010000000,-35.0337010000000,-
35.1548190000000,-35.2701380000000,-35.3798910000000,-35.4842730000000,-
35.5834440000000,-35.6775410000000,-35.7666820000000,-35.8509710000000,-
35.9305060000000,-36.0053800000000,-36.0756850000000,-36.1415110000000,-
36.2029520000000,-36.2601040000000,-36.3130640000000,-36.3619320000000,-
36.4068100000000,-36.4478020000000,-36.4850120000000,-36.5185460000000,-
36.5485100000000,-36.5750080000000,-36.5981460000000,-36.6180270000000,-
36.6347530000000,-36.6484240000000,-36.6591410000000,-36.6669990000000,-
36.6720940000000,-36.6745190000000,-36.6743650000000,-36.6717220000000,-
36.6666760000000,-36.6593130000000,-36.6497160000000,-36.6379650000000,-
36.6241400000000,-36.6083180000000,-36.5905750000000,-36.5709820000000,-
36.5496130000000,-36.5265360000000,-36.5018200000000,-36.4755310000000,-
36.4477320000000,-36.4184870000000,-36.3878560000000,-36.3558990000000,-
36.3226730000000,-36.2882330000000,-36.2526360000000,-36.2159320000000,-
36.1781740000000,-36.1394110000000,-36.0996920000000,-36.0590630000000,-
36.0175700000000,-35.9752570000000,-35.9321660000000,-35.8883400000000,-
35.8438180000000,-35.7986390000000,-35.7528410000000,-35.7064600000000,-
35.6595320000000,-35.6120910000000,-35.5641700000000,-35.5158010000000,-
35.4670140000000,-35.4178410000000,-35.3683090000000,-35.3184470000000,-
35.2682820000000,-35.2178400000000,-35.1671460000000,-35.1162250000000,-
35.0651000000000,-35.0137940000000,-34.9623290000000,-34.9107260000000,-
34.8590060000000,-34.8071870000000,-34.7552900000000,-34.7033320000000,-
34.6513310000000,-34.5993040000000,-34.5472680000000,-34.4952380000000,-
34.4432290000000,-34.3912550000000,-34.3393320000000,-34.2874720000000,-
34.2356880000000,-34.1839930000000,-34.1323980000000,-34.0809150000000,-
34.0295560000000,-33.9783300000000,-33.9272480000000,-33.8763190000000,-
33.8255530000000,-33.7749580000000,-33.7245430000000,-33.6743160000000,-
33.6242850000000,-33.5744570000000,-33.5248400000000,-33.4754390000000,-
33.4262610000000,-33.3773130000000,-33.3286010000000,-33.2801280000000,-
33.2319020000000,-33.1839260000000,-33.1362060000000,-33.0887460000000,-
33.0415500000000,-32.9946210000000,-32.9479650000000,-32.9015830000000,-
32.8554790000000,-32.8096570000000,-32.7641190000000,-32.7188680000000,-
32.6739060000000,-32.6292360000000,-32.5848580000000,-32.5407770000000,-
32.4969920000000,-32.4535060000000,-32.4103210000000,-32.3674360000000,-
32.3248540000000,-32.2825760000000,-32.2406020000000,-32.1989320000000,-
32.1575690000000,-32.1165110000000,-32.0757590000000,-32.0353140000000,-
31.9951760000000,-31.9553440000000,-31.9158190000000,-31.8766000000000,-
31.8376870000000,-31.7990800000000,-31.7607770000000,-31.7227790000000,-
31.6850850000000,-31.6476940000000,-31.6106060000000,-31.5738180000000,-
31.5373320000000,-31.5011440000000,-31.4652550000000,-31.4296630000000,-
31.3943670000000,-31.3593660000000,-31.3246580000000,-31.2902410000000,-
31.2561160000000,-31.2222790000000,-31.1887290000000,-31.1554660000000,-
31.1224870000000,-31.0897900000000,-31.0573740000000,-31.0252370000000,-
30.9933770000000,-30.9617920000000,-30.9304820000000,-30.8994430000000,-
```

30.8686730000000,-30.8381720000000,-30.8079360000000,-30.7779640000000,-
30.7482530000000,-30.7188030000000,-30.6896100000000,-30.6606720000000,-
30.6319880000000,-30.6035550000000,-30.5753720000000,-30.5474350000000,-
30.5197430000000,-30.4922930000000,-30.4650840000000,-30.4381130000000,-
30.4113770000000,-30.3848750000000,-30.3586050000000,-30.3325630000000,-
30.3067480000000,-30.2811580000000,-30.2557890000000,-30.2306410000000,-
30.2057100000000,-30.1809940000000,-30.1564910000000,-30.1321980000000,-
30.1081140000000,-30.0842360000000,-30.0605610000000,-30.0370870000000,-
30.0138120000000,-29.9907340000000,-29.9678490000000,-29.9451570000000,-
29.9226540000000,-29.9003380000000,-29.8782080000000,-29.8562590000000,-
29.8344910000000,-29.8129010000000,-29.7914860000000,-29.7702440000000,-
29.7491730000000,-29.7282710000000,-29.7075350000000,-29.6869630000000,-
29.6665530000000,-29.6463020000000,-29.6262080000000,-29.6062690000000,-
29.5864820000000,-29.5668460000000,-29.5473580000000,-29.5280150000000,-
29.5088160000000,-29.4897590000000,-29.4708400000000,-29.4520580000000,-
29.4334110000000,-29.4148970000000,-29.3965120000000,-29.3782560000000,-
29.3601250000000,-29.3421190000000,-29.3242330000000,-29.3064680000000,-
29.2888190000000,-29.2712860000000,-29.2538650000000,-29.2365550000000,-
29.2193550000000,-29.2022610000000,-29.1852710000000,-29.1683840000000,-
29.1515980000000,-29.1349100000000,-29.1183180000000,-29.1018210000000,-
29.0854160000000,-29.0691020000000,-29.0528750000000,-29.0367350000000,-
29.0206800000000,-29.0047070000000,-28.9888140000000,-28.9730000000000,-
28.9572620000000,-28.9415990000000,-28.9260090000000,-28.9104900000000,-
28.8950390000000,-28.8796560000000,-28.8643380000000,-28.8490830000000,-
28.8338890000000,-28.8187550000000,-28.8036790000000,-28.7886590000000,-
28.7736940000000,-28.7587800000000,-28.7439170000000,-28.7291040000000,-
28.7143370000000,-28.6996150000000,-28.6849370000000,-28.6703010000000,-
28.6557060000000,-28.6411480000000,-28.6266280000000,-28.6121420000000,-
28.5976900000000,-28.5832700000000,-28.5688790000000,-28.5545180000000,-
28.5401830000000,-28.5258730000000,-28.5115860000000,-28.4973220000000,-
28.4830780000000,-28.4688540000000,-28.4546460000000,-28.4404540000000,-
28.4262760000000,-28.4121110000000,-28.3979570000000,-28.3838120000000,-
28.3696760000000,-28.3555460000000,-28.3414210000000,-28.3273000000000,-
28.3131810000000,-28.2990630000000,-28.2849440000000,-28.2708230000000,-
28.2566980000000,-28.2425690000000,-28.2284330000000,-28.2142890000000,-
28.2001360000000,-28.1859720000000,-28.1717970000000,-28.1576080000000,-
28.1434040000000,-28.1291850000000,-28.1149480000000,-28.1006930000000,-
28.0864180000000,-28.0721210000000,-28.0578020000000,-28.0434600000000,-
28.0290920000000,-28.0146980000000,-28.0002760000000,-27.9858250000000,-
27.9713440000000,-27.9568320000000,-27.9422880000000,-27.9277100000000,-
27.9130970000000,-27.8984470000000,-27.8837610000000,-27.8690360000000,-
27.8542710000000,-27.8394650000000,-27.8246180000000,-27.8097270000000,-
27.7947920000000,-27.7798120000000,-27.7647850000000,-27.7497100000000,-
27.7345870000000,-27.7194140000000,-27.7041900000000,-27.6889140000000,-
27.6735850000000,-27.6582020000000,-27.6427630000000,-27.6272690000000,-
27.6117170000000,-27.5961060000000,-27.5804370000000,-27.5647070000000,-
27.5489150000000,-27.5330610000000,-27.5171440000000,-27.5011620000000,-
27.4851150000000,-27.4690020000000,-27.4528210000000,-27.4365720000000,-
27.4202530000000,-27.4038640000000,-27.3874040000000,-27.3708720000000,-
27.3542670000000,-27.3375880000000,-27.3208340000000,-27.3040040000000,-
27.2870970000000,-27.2701130000000,-27.2530490000000,-27.2359070000000,-
27.2186840000000,-27.2013800000000,-27.1839940000000,-27.1665240000000,-
27.1489710000000,-27.1313330000000,-27.1136100000000,-27.0958000000000,-
27.0779030000000,-27.0599170000000,-27.0418430000000,-27.0236790000000,-
27.0054240000000,-26.9870780000000,-26.9686390000000,-26.9501080000000,-
26.9314820000000,-26.9127620000000,-26.8939460000000,-26.8750350000000,-
26.8560260000000,-26.8369190000000,-26.8177130000000,-26.7984080000000,-

26.7790030000000,-26.7594970000000,-26.7398890000000,-26.7201790000000,-
26.7003650000000,-26.6804480000000,-26.6604260000000,-26.6402980000000,-
26.6200640000000,-26.5997230000000,-26.5792740000000,-26.5587170000000,-
26.5380500000000,-26.5172740000000,-26.4963870000000,-26.4753890000000,-
26.4542790000000,-26.4330550000000,-26.4117190000000,-26.3902680000000,-
26.3687020000000,-26.3470210000000,-26.3252230000000,-26.3033080000000,-
26.2812760000000,-26.2591250000000,-26.2368550000000,-26.2144660000000,-
26.1919550000000,-26.1693240000000,-26.1465710000000,-26.1236950000000,-
26.1006970000000,-26.0775740000000,-26.0543260000000,-26.0309540000000,-
26.0074560000000,-25.9838310000000,-25.9600780000000,-25.9361980000000,-
25.9121900000000,-25.8880520000000,-25.8637840000000,-25.8393860000000,-
25.8148560000000,-25.7901950000000,-25.7654010000000,-25.7404740000000,-
25.7154130000000,-25.6902180000000,-25.6648880000000,-25.6394210000000,-
25.6138190000000,-25.5880790000000,-25.5622010000000,-25.5361850000000,-
25.5100300000000,-25.4837350000000,-25.4573000000000,-25.4307240000000,-
25.4040070000000,-25.3771470000000,-25.3501440000000,-25.3229970000000,-
25.2957070000000,-25.2682710000000,-25.2406900000000,-25.2129630000000,-
25.1850890000000,-25.1570670000000,-25.1288980000000,-25.1005790000000,-
25.0721110000000,-25.0434940000000,-25.0147250000000,-24.9858050000000,-
24.9567330000000,-24.9275080000000,-24.8981310000000,-24.8685990000000,-
24.8389120000000,-24.8090700000000,-24.7790730000000,-24.7489190000000,-
24.7186070000000,-24.6881380000000,-24.6575100000000,-24.6267230000000,-
24.5957760000000,-24.5646680000000,-24.5334000000000,-24.5019690000000,-
24.4703760000000,-24.4386200000000,-24.4067000000000,-24.3746150000000,-
24.3423650000000,-24.3099490000000,-24.2773670000000,-24.2446180000000,-
24.2117000000000,-24.1786150000000,-24.1453590000000,-24.1119340000000,-
24.0783390000000,-24.0445720000000,-24.0106330000000,-23.9765210000000,-
23.9422360000000,-23.9077770000000,-23.8731430000000,-23.8383340000000,-
23.8033490000000,-23.7681860000000,-23.7328460000000,-23.6973280000000,-
23.6616310000000,-23.6257550000000,-23.5896980000000,-23.5534590000000,-
23.5170390000000,-23.4804370000000,-23.4436510000000,-23.4066810000000,-
23.3695270000000,-23.3321870000000,-23.2946610000000,-23.2569480000000,-
23.2190470000000,-23.1809580000000,-23.1426800000000,-23.1042120000000,-
23.0655540000000,-23.0267040000000,-22.9876620000000,-22.9484270000000,-
22.9089980000000,-22.8693750000000,-22.8295570000000,-22.7895440000000,-
22.7493330000000,-22.7089250000000,-22.6683190000000,-22.6275130000000,-
22.5865080000000,-22.5453030000000,-22.5038960000000,-22.4622870000000,-
22.4204750000000,-22.3784590000000,-22.3362390000000,-22.2938130000000,-
22.2511810000000,-22.2083430000000,-22.1652960000000,-22.1220410000000,-
22.0785770000000,-22.0349020000000,-21.9910170000000,-21.9469190000000,-
21.9026090000000,-21.8580850000000,-21.8133470000000,-21.7683940000000,-
21.7232250000000,-21.6778390000000,-21.6322350000000,-21.5864120000000,-
21.5403700000000,-21.4941080000000,-21.4476240000000,-21.4009180000000,-
21.3539900000000,-21.3068370000000,-21.2594600000000,-21.2118580000000,-
21.1640290000000,-21.1159720000000,-21.0676870000000,-21.0191740000000,-
20.9704300000000,-20.9214550000000,-20.8722480000000,-20.8228090000000,-
20.7731360000000,-20.7232280000000,-20.6730850000000,-20.6227050000000,-
20.5720880000000,-20.5212330000000,-20.4701390000000,-20.4188040000000,-
20.3672280000000,-20.3154100000000,-20.2633490000000,-20.2110440000000,-
20.1584950000000,-20.1056990000000,-20.0526560000000,-19.9993660000000,-
19.9458270000000,-19.8920380000000,-19.8379980000000,-19.7837060000000,-
19.7291620000000,-19.6743640000000,-19.6193120000000,-19.5640030000000,-
19.5084380000000,-19.4526160000000,-19.3965340000000,-19.3401930000000,-
19.2835910000000,-19.2267280000000,-19.1696010000000,-19.1122110000000,-
19.0545560000000,-18.9966350000000,-18.9384470000000,-18.8799910000000,-
18.8212670000000,-18.7622720000000,-18.7030060000000,-18.6434680000000,-
18.5836570000000,-18.5235720000000,-18.4632120000000,-18.4025750000000,-

18.3416610000000,-18.2804680000000,-18.2189960000000,-18.1572430000000,-
18.0952080000000,-18.0328910000000,-17.9702900000000,-17.9074040000000,-
17.8442320000000,-17.7807730000000,-17.7170260000000,-17.6529890000000,-
17.5886620000000,-17.5240440000000,-17.4591330000000,-17.3939280000000,-
17.3284290000000,-17.2626340000000,-17.1965410000000,-17.1301510000000,-
17.0634620000000,-16.9964720000000,-16.9291810000000,-16.8615870000000,-
16.7936900000000,-16.7254880000000,-16.6569800000000,-16.5881650000000,-
16.5190420000000,-16.4496100000000,-16.3798680000000,-16.3098140000000,-
16.2394470000000,-16.1687670000000,-16.0977720000000,-16.0264610000000,-
15.9548340000000,-15.8828870000000,-15.8106220000000,-15.7380360000000,-
15.6651290000000,-15.5918990000000,-15.5183460000000,-15.4444670000000,-
15.3702630000000,-15.2957310000000,-15.2208710000000,-15.1456830000000,-
15.0701630000000,-14.9943120000000,-14.9181290000000,-14.8416120000000,-
14.7647600000000,-14.6875720000000,-14.6100470000000,-14.5321850000000,-
14.4539830000000,-14.3754410000000,-14.2965570000000,-14.2173320000000,-
14.1377620000000,-14.0578490000000,-13.9775900000000,-13.8969840000000,-
13.8160310000000,-13.7347290000000,-13.6530770000000,-13.5710750000000,-
13.4887210000000,-13.4060140000000,-13.3229540000000,-13.2395380000000,-
13.1557670000000,-13.0716400000000,-12.9871540000000,-12.9023100000000,-
12.8171070000000,-12.7315420000000,-12.6456170000000,-12.5593290000000,-
12.4726770000000,-12.3856610000000,-12.2982800000000,-12.2105330000000,-
12.1224190000000,-12.0339370000000,-11.9450870000000,-11.8558670000000,-
11.7662770000000,-11.6763150000000,-11.5859820000000,-11.4952760000000,-
11.4041960000000,-11.3127420000000,-11.2209130000000,-11.1287080000000,-
11.0361270000000,-10.9431680000000,-10.8498320000000,-10.7561170000000,-
10.6620230000000,-10.5675490000000,-10.4726940000000,-10.3774590000000,-
10.2818410000000,-10.1858420000000,-10.0894600000000,-9.99269400000000,-
9.89554500000000,-9.79801100000000,-9.70009300000000,-9.60178900000000,-
9.50310000000000,-9.40402500000000,-9.30456300000000,-9.20471500000000,-
9.10448000000000,-9.00385700000000,-8.90284700000000,-8.80144800000000,-
8.69966200000000,-8.59748700000000,-8.49492400000000,-8.39197200000000,-
8.28863200000000,-8.18490200000000,-8.08078400000000,-7.97627600000000,-
7.87138000000000,-7.76609400000000,-7.66042000000000,-7.55435700000000,-
7.44790500000000,-7.34106500000000,-7.23383600000000,-7.12621900000000,-
7.01821400000000,-6.90982100000000,-6.80104200000000,-6.69187500000000,-
6.58232100000000,-6.47238200000000,-6.36205600000000,-6.25134500000000,-
6.14025000000000,-6.02877000000000,-5.91690700000000,-5.80466100000000,-
5.69203300000000,-5.57902300000000,-5.46563300000000,-5.35186200000000,-
5.23771300000000,-5.12318500000000,-5.00828000000000,-4.89299900000000,-
4.77734200000000,-4.66131100000000,-4.54490700000000,-4.42813100000000,-
4.31098400000000,-4.19346800000000,-4.07558300000000,-3.95733200000000,-
3.83871500000000,-3.71973300000000,-3.60038900000000,-3.48068400000000,-
3.36061900000000,-3.24019700000000,-3.11941800000000,-2.99828400000000,-
2.87679800000000,-2.75496000000000,-2.63277400000000,-2.51024000000000,-
2.38736100000000,-2.26413800000000,-2.14057500000000,-2.01667200000000,-
1.89243300000000,-1.76785900000000,-1.64295300000000,-1.51771700000000,-
1.39215300000000,-1.26626400000000,-1.14005300000000,-1.01352100000000,-
0.886673000000000,-0.759509000000000,-0.632033000000000,-
0.504249000000000,-0.376158000000000,-0.247763000000000,-
0.119068000000000,0.00992400000000000,0.139211000000000,0.268789000000000,0
.398655000000000,0.528806000000000,0.659239000000000,0.789949000000000,0.92
0934000000000,1.05219000000000,1.18371400000000,1.31550200000000,1.44755100
000000,1.57985500000000,1.71241300000000,1.84522000000000,1.97827200000000,
2.11156500000000,2.24509500000000,2.37885800000000,2.51285000000000,2.64706
700000000,2.78150400000000,2.91615700000000,3.05102200000000,3.186094000000
00,3.32136900000000,3.45684300000000,3.59250900000000,3.72836500000000,3.86
440500000000,4.00062500000000,4.13701800000000,4.27358200000000,4.410310000

```
00000,4.54719800000000,4.68424100000000,4.82143300000000,4.95876900000000,5
.09624400000000,5.23385300000000,5.37159100000000,5.50945100000000,5.647429
00000000,5.78551800000000,5.92371400000000,6.06201100000000,6.2004030000000
0,6.33888400000000,6.47744800000000,6.61609000000000,6.75480400000000,6.893
58300000000,7.03242300000000,7.17131600000000,7.31025600000000,7.4492380000
0000,7.58825600000000,7.72730200000000,7.86637000000000,8.00545500000000,8.
14455000000000,8.28364800000000,8.42274300000000,8.56182900000000,8.7008980
0000000,8.83994300000000,8.97896000000000,9.11793900000000,9.25687600000000
,9.39576200000000,9.53459100000000,9.67335600000000,9.81204900000000,9.9506
6500000000,10.0891950000000,10.2276320000000,10.3659690000000,10.5042000000
000,10.6423150000000,10.7803090000000,10.9181720000000,11.0558990000000,11.
1934800000000,11.3309080000000,11.4681760000000,11.6052750000000,11.7421970
000000,11.8789330000000,12.0154760000000,12.1518170000000,12.2879470000000,
12.4238570000000,12.5595380000000,12.6949800000000,12.8301740000000,12.9651
100000000,13.0997780000000,13.2341650000000,13.3682620000000,13.50205600000
00,13.6355350000000,13.7686840000000,13.9014910000000,14.0339380000000,14.1
660100000000,14.2976870000000,14.4289490000000,14.5597740000000,14.69013400
00000,14.8200020000000,14.9493430000000,15.0781160000000,15.2062760000000,1
5.3337670000000,15.4605210000000,15.5864560000000,15.7114680000000,15.83542
80000000,15.9581700000000,16.0794790000000,16.1990680000000,16.316551000000
0,16.4313960000000,16.5428490000000,16.6498140000000,16.7506480000000,16.84
27880000000,16.9220600000000,16.9812710000000,17.0071300000000,16.972718000
0000,16.8160680000000,16.3642710000000,14.9389210000000,4.82986100000000,37
.7182530000000];
% webot_torque = -A(:,4);
% plot(T, webot_torque, '--k')
% CTC_torque = -B(:,4);
% plot(T, CTC_torque, '--r')

%% Newton

% for k = 1:1001;
%
%    R32x = 0.185*cos(I_K_th1(k));
%    R32y = 0.185*sin(I_K_th1(k));
%    R23x = -0.219*cos(I_K_th1(k) + I_K_th2_star(k));
%    R23y = -0.219*sin(I_K_th1(k) + I_K_th2_star(k));
%    R43x = 0.219*cos(I_K_th1(k) + I_K_th2_star(k));
%    R43y = 0.219*sin(I_K_th1(k) + I_K_th2_star(k));
%    R53x = 0.219*cos(I_K_th1(k) + I_K_th2_star(k));
%    R53y = 0.219*sin(I_K_th1(k) + I_K_th2_star(k));
%    R34x = -0.227*cos(I_K_th1(k) + I_K_th2_star(k) + I_K_th4_star(k));
%    R34y = -0.227*sin(I_K_th1(k) + I_K_th2_star(k) + I_K_th4_star(k));
%    R14x = 0.227*cos(I_K_th1(k) + I_K_th2_star(k) + I_K_th4_star(k));
%    R14y = 0.227*sin(I_K_th1(k) + I_K_th2_star(k) + I_K_th4_star(k));
%
%
%    m2 = 7.7;
%    m3 = 1.7;
%    m4 = 2.4;
%    m5 = 0.8;
%
%    xi = [1;0;0];
%    yj = [0;1;0];
%    zk = [0;0;1];
%
%    a2gx = 0;
```

```
%     a2gy = 0;
%     a2ge = cross(I_K_th1_dotdot(k)*zk, L1*xi) + cross(I_K_th1_dot(k)*zk,
cross(I_K_th1_dot(k)*zk, L1*xi));
%
%     a3gc = R2'*a2ge + cross(I_K_th2_dotdot(k)*zk, L2/2*xi) +
cross(I_K_th2_dot(k)*zk, cross(I_K_th2_dot(k)*zk, L2/2*xi));
%     a3gx = a3gc(1,:);
%     a3gy = a3gc(2,:);
%     a3ge = R2'*a2ge + cross(I_K_th2_dotdot(k)*zk, L2*xi) +
cross(I_K_th2_dot(k)*zk, cross(I_K_th2_dot(k)*zk, L2*xi));
%
%     a4gc = R4'*a3ge + cross(I_K_th4_dotdot(k)*zk, L4/2*xi) +
cross(I_K_th4_dot(k)*zk, cross(I_K_th4_dot(k)*zk, L4/2*xi));
%     a4gx = a4gc(1,:);
%     a4gy = a4gc(2,:);
%     a4ge = R4'*a3ge + cross(I_K_th4_dotdot(k)*zk, L4*xi) +
cross(I_K_th4_dot(k)*zk, cross(I_K_th4_dot(k)*zk, L4*xi));
%
%     a5gc = R3'*a3ge;
%     a5gx = a5gc(1,:);
%     a5gy = a5gc(2,:);
%     a5ge = R3'*a3ge;
%
%     a2 = 0;
%     a3 = I_K_th2_dotdot(k);
%     a4 = I_K_th4_dotdot(k);
%
%     Ig2 = m2*L1^2 * 10^-6;
%     Ig3 = m3*(0.05^2 + L2^2) / 12 * 10^-6;
%     Ig4 = m4*(L4^2 + 0.05^2) / 12 * 10^-6;
%     Ig5 = m5*(L3^2 + 0.05^2) / 12 * 10^-6;
%
%     Fpx = 0;
%     Fpy = 0;
%
%     A = [1 0 1 0 0 0 0 0 0 0 0
%         0 1 0 1 0 0 0 0 0 0 0
%         0 0 -R32y R32x 0 0 0 0 0 0 1
%         0 0 -1 0 1 0 1 0 0 0 0
%         0 0 0 -1 0 1 0 1 0 0 0
%         0 0 R23y -R23x -R43y R43x -R53y R53x 0 0 0
%         0 0 0 0 -1 0 0 0 1 0 0
%         0 0 0 0 0 -1 0 0 0 1 0
%         0 0 0 0 R34y -R34x 0 0 -R14y R14x 0
%         0 0 0 0 0 0 -1 0 0 0 0
%         0 0 0 0 0 0 0 -1 0 0 0];
%
%     B(:,k) = [m2*a2gx; m2*a2gy; Ig2*a2; m3*a3gx; m3*a3gy; Ig3*a3; m4*a4gx;
m4*a4gy; Ig4*a4; m5*a5gx-Fpx; m5*a5gy-Fpy];
%
%     C(:,k) = inv(A) * B(:, k); % F12x, F12y, F32x, F32y, F43x, F43y, F53x,
F53y, F14x, F14y, Tm
% end
%
% % I_A = inv(A);
% %
% % torque_N = C(11,:);
```

```
% %
% % plot(T,torque_N)
%
```

5.graph.m

```
%%%%%%%%%%%°¢µµ¿ë
clear all;%% close all; clc;
fname = 'C:\Users\whgdm\OneDrive\¹ÙÅÁ È-¸é\¼ö¾÷\1.
·Î°¿µ¿¿ªÇÐ¹×Á¦¾î\webot\controllers\4Link\data2.txt';
A = load(fname);

scale = 1;
% scale = 180/pi;

x = size(A);
buf_exw1 = -A(:,1)*180/pi;
buf_exw2 = -A(:,2)*180/pi;
buf_exw3 = -A(:,3)*scale;
buf_exw4 = -A(:,4)*scale;
buf_exw5 = -A(:,5);

fig15 = figure(5);
fig15.Color = 'white';
fig15.Position = ([100 100 2000 800]);

dt = 0.001;
freq = 1/dt;

ylabel("Torque(Nm)")
xlabel("Time [sec]")
title('Right Arm Torque');
grid on
hold on
plot(dt:dt:(x(1))/freq,buf_exw1,'LineWidth',1);
plot(dt:dt:(x(1))/freq,buf_exw2,'LineWidth',1);
% plot(dt:dt:(x(1))/freq,buf_exw3,'LineWidth',1);
plot(dt:dt:(x(1))/freq,buf_exw4,'LineWidth',1);
% plot(dt:dt:(x(1))/freq,buf_exw5,'LineWidth',1);
```

B. Mathemetica

```
Jvc1=({
   {-L1*Sin[th1]-L2*Sin[th1+th2]/2},
   {L1*Cos[th1]+L2*Cos[th1+th2]/2},
   {0}
  })
{{-L1 Sin[th1]-1/2 L2 Sin[th1+th2]},{L1 Cos[th1]+1/2 L2 Cos[th1+th2]},{0}}
Jvc2=({
   {-L1*Sin[th1]-L2*Sin[th1+th2]-L3*Sin[th1+th2+th3]/2},
   {L1*Cos[th1]+L2*Cos[th1+th2]+L3*Cos[th1+th2+th3]/2},
```

```
    {0}
   })
{{-L1 Sin[th1]-L2 Sin[th1+th2]-1/2 L3 Sin[th1+th2+th3]},{L1 Cos[th1]+L2
Cos[th1+th2]+1/2 L3 Cos[th1+th2+th3]},{0}}
TJvc1=Transpose[Jvc1]
{{-L1 Sin[th1]-1/2 L2 Sin[th1+th2],L1 Cos[th1]+1/2 L2 Cos[th1+th2],0}}
TJvc2=Transpose[Jvc2]
{{-L1 Sin[th1]-L2 Sin[th1+th2]-1/2 L3 Sin[th1+th2+th3],L1 Cos[th1]+L2 Cos[th1+th2]+1/2
L3 Cos[th1+th2+th3],0}}
Rc1=({
    {Cos[th1+th2], -Sin[th1+th2], 0},
    {Sin[th1+th2], Cos[th1+th2], 0},
    {0, 0, 1}
   })
{{Cos[th1+th2],-Sin[th1+th2],0},{Sin[th1+th2],Cos[th1+th2],0},{0,0,1}}
TRc1=Transpose[Rc1]
{{Cos[th1+th2],Sin[th1+th2],0},{-Sin[th1+th2],Cos[th1+th2],0},{0,0,1}}
Rc2=({
    {Cos[th1+th2+th3], -Sin[th1+th2+th3], 0},
    {Sin[th1+th2+th3], Cos[th1+th2+th3], 0},
    {0, 0, 1}
   })
{{Cos[th1+th2+th3],-Sin[th1+th2+th3],0},{Sin[th1+th2+th3],Cos[th1+th2+th3],0},{0,0,1}}
TRc2=Transpose[Rc2]
{{Cos[th1+th2+th3],Sin[th1+th2+th3],0},{-Sin[th1+th2+th3],Cos[th1+th2+th3],0},{0,0,1}}
Ic1=({
    {Ic1xx, 0, 0},
    {0, Ic1yy, 0},
    {0, 0, Ic1zz}
   })
{{Ic1xx,0,0},{0,Ic1yy,0},{0,0,Ic1zz}}
Ic2=({
    {Ic2xx, 0, 0},
    {0, Ic2yy, 0},
    {0, 0, Ic2zz}
   })
{{Ic2xx,0,0},{0,Ic2yy,0},{0,0,Ic2zz}}
Jwc1=({
    {0},
    {0},
    {1}
   })
{{0},{0},{1}}
Jwc2=({
    {0},
    {0},
    {1}
```

})
{{0},{0},{1}}

TJwc1=Transpose[Jwc1]
{{0,0,1}}
TJwc2=Transpose[Jwc2]
{{0,0,1}}
m1*TJvc1.Jvc1+m2*TJvc2.Jvc2+TJwc1.Rc1.Ic1.TRc1.Jwc1+TJwc2.Rc2.Ic2.TRc2.Jwc2
{{Ic1zz+Ic2zz+m1 ((L1 Cos[th1]+1/2 L2 Cos[th1+th2])^2+(-L1 Sin[th1]-1/2 L2 Sin[th1+th2])^2)+m2 ((L1 Cos[th1]+L2 Cos[th1+th2]+1/2 L3 Cos[th1+th2+th3])^2+(-L1 Sin[th1]-L2 Sin[th1+th2]-1/2 L3 Sin[th1+th2+th3])^2)}}
FullSimplify[MatrixForm[m1*TJvc1.Jvc1+m2*TJvc2.Jvc2+TJwc1.Rc1.Ic1.TRc1.Jwc1+TJwc2.Rc2.Ic2.TRc2.Jwc2]]
({
{Ic1zz+Ic2zz+L1^2 (m1+m2)+1/4 (L3^2 m2+L2^2 (m1+4 m2))+L1 L2 (m1+2 m2) Cos[th2]+L3 m2 (L2 Cos[th3]+L1 Cos[th2+th3])}
})
FullSimplify[m1*TJvc1.Jvc1+m2*TJvc2.Jvc2+TJwc1.Rc1.Ic1.TRc1.Jwc1+TJwc2.Rc2.Ic2.TRc2.Jwc2]
{{Ic1zz+Ic2zz+L1^2 (m1+m2)+1/4 (L3^2 m2+L2^2 (m1+4 m2))+L1 L2 (m1+2 m2) Cos[th2]+L3 m2 (L2 Cos[th3]+L1 Cos[th2+th3])}}
d11=m1*TJvc1.Jvc1+m2*TJvc2.Jvc2+TJwc1.Rc1.Ic1.TRc1.Jwc1+TJwc2.Rc2.Ic2.TRc2.Jwc2
{{Ic1zz+Ic2zz+m1 ((L1 Cos[th1]+1/2 L2 Cos[th1+th2])^2+(-L1 Sin[th1]-1/2 L2 Sin[th1+th2])^2)+m2 ((L1 Cos[th1]+L2 Cos[th1+th2]+1/2 L3 Cos[th1+th2+th3])^2+(-L1 Sin[th1]-L2 Sin[th1+th2]-1/2 L3 Sin[th1+th2+th3])^2)}}
c111=FullSimplify[(1/2)*(D[d11,th1]+D[d11,th1]-D[d11,th1])]
{{0}}

P=m1*g*(L1*Sin[th1]+L2*Sin[th1+th2]/2)+m2*g*(L1*Sin[th1]+L2*Sin[th1+th2]+L3*Sin[th1+th2+th3]/2)
g m1 (L1 Sin[th1]+1/2 L2 Sin[th1+th2])+g m2 (L1 Sin[th1]+L2 Sin[th1+th2]+1/2 L3 Sin[th1+th2+th3])
g1=D[P,th1]
g m1 (L1 Cos[th1]+1/2 L2 Cos[th1+th2])+g m2 (L1 Cos[th1]+L2 Cos[th1+th2]+1/2 L3 Cos[th1+th2+th3])
t=d11+g1
{{Ic1zz+Ic2zz+g m1 (L1 Cos[th1]+1/2 L2 Cos[th1+th2])+g m2 (L1 Cos[th1]+L2 Cos[th1+th2]+1/2 L3 Cos[th1+th2+th3])+m1 ((L1 Cos[th1]+1/2 L2 Cos[th1+th2])^2+(-L1 Sin[th1]-1/2 L2 Sin[th1+th2])^2)+m2 ((L1 Cos[th1]+L2 Cos[th1+th2]+1/2 L3 Cos[th1+th2+th3])^2+(-L1 Sin[th1]-L2 Sin[th1+th2]-1/2 L3 Sin[th1+th2+th3])^2)}}
Jvc3=({
{-(330 +L1*Sin[th1]+L2*Sin[th1+th2])/2},
{(547.5+L1*Cos[th1]+L2*Cos[th1+th2])/2},
{0}
})
{{1/2 (-330-L1 Sin[th1]-L2 Sin[th1+th2])},{1/2 (547.5 +L1 Cos[th1]+L2

```
Cos[th1+th2])},{0}}
TJvc3=Transpose[Jvc3]
{{1/2 (-330-L1 Sin[th1]-L2 Sin[th1+th2]),1/2 (547.5 +L1 Cos[th1]+L2 Cos[th1+th2]),0}}
{{1/2 (-330-L1 Sin[th1]-L2 Sin[th1+th2]),1/2 (547.5` +L1 Cos[th1]+L2 Cos[th1+th2]),0}}
{{1/2 (-330-L1 Sin[th1]-L2 Sin[th1+th2]),1/2 (547.5 +L1 Cos[th1]+L2 Cos[th1+th2]),0}}
Jvc1=({
    {-L1*Sin[th1]-L2*Sin[th1+th2]/2},
    {L1*Cos[th1]+L2*Cos[th1+th2]/2},
    {0}
  })
{{-L1 Sin[th1]-1/2 L2 Sin[th1+th2]},{L1 Cos[th1]+1/2 L2 Cos[th1+th2]},{0}}
Jvc2=({
    {-L1*Sin[th1]-L2*Sin[th1+th2]-L3*Sin[th1+th2+th3]/2},
    {L1*Cos[th1]+L2*Cos[th1+th2]+L3*Cos[th1+th2+th3]/2},
    {0}
  })
{{-L1 Sin[th1]-L2 Sin[th1+th2]-1/2 L3 Sin[th1+th2+th3]},{L1 Cos[th1]+L2
Cos[th1+th2]+1/2 L3 Cos[th1+th2+th3]},{0}}
Rc3=({
    {Cos[th1+th2+th4], -Sin[th1+th2+th4], 0},
    {Sin[th1+th2+th4], Cos[th1+th2+th4], 0},
    {0, 0, 1}
  })
{{Cos[th1+th2+th4],-Sin[th1+th2+th4],0},{Sin[th1+th2+th4],Cos[th1+th2+th4],0},{0,0,1}}
Ic3=({
    {Ic3xx, 0, 0},
    {0, Ic3yy, 0},
    {0, 0, Ic3zz}
  })
{{Ic3xx,0,0},{0,Ic3yy,0},{0,0,Ic3zz}}
TRc3=Transpose[Rc3]
{{Cos[th1+th2+th4],Sin[th1+th2+th4],0},{-Sin[th1+th2+th4],Cos[th1+th2+th4],0},{0,0,1}}
Jwc3=({
    {0},
    {0},
    {1}
  })
{{0},{0},{1}}
TJwc3=Transpose[Jwc2]
{{0,0,1}}
d11=m1*TJvc1.Jvc1+m2*TJvc2.Jvc2+m3*TJvc3.Jvc3+TJwc1.Rc1.Ic1.TRc1.Jwc1+TJwc2.
Rc2.Ic2.TRc2.Jwc2+TJwc3.Rc3.Ic3.TRc3.Jwc3
{{Ic1zz+Ic2zz+Ic3zz+m3 (1/4 (547.5 +L1 Cos[th1]+L2 Cos[th1+th2])^2+1/4 (-330-L1
Sin[th1]-L2 Sin[th1+th2])^2)+m1 ((L1 Cos[th1]+1/2 L2 Cos[th1+th2])^2+(-L1 Sin[th1]-1/2 L2
Sin[th1+th2])^2)+m2 ((L1 Cos[th1]+L2 Cos[th1+th2]+1/2 L3 Cos[th1+th2+th3])^2+(-L1
Sin[th1]-L2 Sin[th1+th2]-1/2 L3 Sin[th1+th2+th3])^2)}}
P=m1*g*(L1*Sin[th1]+L2*Sin[th1+th2]/2)+m2*g*(L1*Sin[th1]+L2*Sin[th1+th2]+L3*Sin[t
```

h1+th2+th3]/2)+m3*g*(L1*Sin[th1]+L2*Sin[th1+th2]+L4*Sin[th1+th2+th4]/2)
g m1 (L1 Sin[th1]+1/2 L2 Sin[th1+th2])+g m2 (L1 Sin[th1]+L2 Sin[th1+th2]+1/2 L3 Sin[th1+th2+th3])+g m3 (L1 Sin[th1]+L2 Sin[th1+th2]+1/2 L4 Sin[th1+th2+th4])

g1=D[P,th1]
g m1 (L1 Cos[th1]+1/2 L2 Cos[th1+th2])+g m2 (L1 Cos[th1]+L2 Cos[th1+th2]+1/2 L3 Cos[th1+th2+th3])+g m3 (L1 Cos[th1]+L2 Cos[th1+th2]+1/2 L4 Cos[th1+th2+th4])
c111=FullSimplify[(1/2)*(D[d11,th1]+D[d11,th1]-D[d11,th1])]
{{82.5 m3 (L1 Cos[th1]+L2 Cos[th1+th2]-1.65909 L1 Sin[th1]-1.65909 L2 Sin[th1+th2])}}
d11=m1*TJvc1.Jvc1+m2*TJvc2.Jvc2+m3*TJvc3.Jvc3+TJwc1.Rc1.Ic1.TRc1.Jwc1+TJwc2.Rc2.Ic2.TRc2.Jwc2+TJwc3.Rc3.Ic3.TRc3.Jwc3
{{Ic1zz+Ic2zz+Ic3zz+m3 (1/4 (547.5 +L1 Cos[th1]+L2 Cos[th1+th2])^{2}+1/4 (-330-L1 Sin[th1]-L2 Sin[th1+th2])^{2})+m1 ((L1 Cos[th1]+1/2 L2 Cos[th1+th2])^{2}+(-L1 Sin[th1]-1/2 L2 Sin[th1+th2])^{2})+m2 ((L1 Cos[th1]+L2 Cos[th1+th2]+1/2 L3 Cos[th1+th2+th3])^{2}+(-L1 Sin[th1]-L2 Sin[th1+th2]-1/2 L3 Sin[th1+th2+th3])^{2})}}
TJwc3.Rc3.Ic3.TRc3.Jwc3
{{Ic3zz}}
TJwc2.Rc2.Ic2.TRc2.Jwc2
{{Ic2zz}}
d11=m1*TJvc1.Jvc1+m2*TJvc2.Jvc2+m3*TJvc3.Jvc3+TJwc1.Rc1.Ic1.TRc1.Jwc1+TJwc2.Rc2.Ic2.TRc2.Jwc2+TJwc3.Rc3.Ic3.TRc3.Jwc3
{{Ic1zz+Ic2zz+Ic3zz+m3 (1/4 (547.5 +L1 Cos[th1]+L2 Cos[th1+th2])^{2}+1/4 (-330-L1 Sin[th1]-L2 Sin[th1+th2])^{2})+m1 ((L1 Cos[th1]+1/2 L2 Cos[th1+th2])^{2}+(-L1 Sin[th1]-1/2 L2 Sin[th1+th2])^{2})+m2 ((L1 Cos[th1]+L2 Cos[th1+th2]+1/2 L3 Cos[th1+th2+th3])^{2}+(-L1 Sin[th1]-L2 Sin[th1+th2]-1/2 L3 Sin[th1+th2+th3])^{2})}}
P=m1*g*(L1*Sin[th1]+L2*Sin[th1+th2]/2)+m2*g*(L1*Sin[th1]+L2*Sin[th1+th2]+L3*Sin[th1+th2+th3]/2)+m3*g*(L1*Sin[th1]+L2*Sin[th1+th2]+L4*Sin[th1+th2+th4]/2)
g m1 (L1 Sin[th1]+1/2 L2 Sin[th1+th2])+g m2 (L1 Sin[th1]+L2 Sin[th1+th2]+1/2 L3 Sin[th1+th2+th3])+g m3 (L1 Sin[th1]+L2 Sin[th1+th2]+1/2 L4 Sin[th1+th2+th4])
c111=FullSimplify[(1/2)*(D[d11,th1]+D[d11,th1]-D[d11,th1])]
{{82.5 m3 (L1 Cos[th1]+L2 Cos[th1+th2]-1.65909 L1 Sin[th1]-1.65909 L2 Sin[th1+th2])}}
t=d11+g1+c111
{{Ic1zz+Ic2zz+Ic3zz+g m1 (L1 Cos[th1]+1/2 L2 Cos[th1+th2])+g m2 (L1 Cos[th1]+L2 Cos[th1+th2]+1/2 L3 Cos[th1+th2+th3])+g m3 (L1 Cos[th1]+L2 Cos[th1+th2]+1/2 L4 Cos[th1+th2+th4])+82.5 m3 (L1 Cos[th1]+L2 Cos[th1+th2]-1.65909 L1 Sin[th1]-1.65909 L2 Sin[th1+th2])+m3 (1/4 (547.5 +L1 Cos[th1]+L2 Cos[th1+th2])^{2}+1/4 (-330-L1 Sin[th1]-L2 Sin[th1+th2])^{2})+m1 ((L1 Cos[th1]+1/2 L2 Cos[th1+th2])^{2}+(-L1 Sin[th1]-1/2 L2 Sin[th1+th2])^{2})+m2 ((L1 Cos[th1]+L2 Cos[th1+th2]+1/2 L3 Cos[th1+th2+th3])^{2}+(-L1 Sin[th1]-L2 Sin[th1+th2]-1/2 L3 Sin[th1+th2+th3])^{2})}}

```
Jvc3=({
    {-(y +L1*Sin[th1]+L2*Sin[th1+th2])/2},
    {(x+L1*Cos[th1]+L2*Cos[th1+th2])/2},
    {0}
  })
```

{{1/2 (-y-L1 Sin[th1]-L2 Sin[th1+th2])},{1/2 (x+L1 Cos[th1]+L2 Cos[th1+th2])},{0}}
{{1/2 (-y-L1 Sin[th1]-L2 Sin[th1+th2])},{1/2 (x+L1 Cos[th1]+L2 Cos[th1+th2])},{0}}

```
{{1/2 (-y-L1 Sin[th1]-L2 Sin[th1+th2])},{1/2 (x+L1 Cos[th1]+L2 Cos[th1+th2])},{0}}
TJvc3=Transpose[Jvc3]
{{1/2 (-y-L1 Sin[th1]-L2 Sin[th1+th2]),1/2 (x+L1 Cos[th1]+L2 Cos[th1+th2]),0}}
d11=m1*TJvc1.Jvc1+m2*TJvc2.Jvc2+m3*TJvc3.Jvc3+TJwc1.Rc1.Ic1.TRc1.Jwc1+TJwc2.
Rc2.Ic2.TRc2.Jwc2+TJwc3.Rc3.Ic3.TRc3.Jwc3
{{Ic1zz+Ic2zz+Ic3zz+m3 (1/4 (x+L1 Cos[th1]+L2 Cos[th1+th2])^2+1/4 (-y-L1 Sin[th1]-L2
Sin[th1+th2])^2)+m1 ((L1 Cos[th1]+1/2 L2 Cos[th1+th2])^2+(-L1 Sin[th1]-1/2 L2
Sin[th1+th2])^2)+m2 ((L1 Cos[th1]+L2 Cos[th1+th2]+1/2 L3 Cos[th1+th2+th3])^2+(-L1
Sin[th1]-L2 Sin[th1+th2]-1/2 L3 Sin[th1+th2+th3])^2)}}
P=m1*g*(L1*Sin[th1]+L2*Sin[th1+th2]/2)+m2*g*(L1*Sin[th1]+L2*Sin[th1+th2]+L3*Sin[t
h1+th2+th3]/2)+m3*g*(L1*Sin[th1]+L2*Sin[th1+th2]+L4*Sin[th1+th2+th4]/2)
g m1 (L1 Sin[th1]+1/2 L2 Sin[th1+th2])+g m2 (L1 Sin[th1]+L2 Sin[th1+th2]+1/2 L3
Sin[th1+th2+th3])+g m3 (L1 Sin[th1]+L2 Sin[th1+th2]+1/2 L4 Sin[th1+th2+th4])
g1=D[P,th1]
g m1 (L1 Cos[th1]+1/2 L2 Cos[th1+th2])+g m2 (L1 Cos[th1]+L2 Cos[th1+th2]+1/2 L3
Cos[th1+th2+th3])+g m3 (L1 Cos[th1]+L2 Cos[th1+th2]+1/2 L4 Cos[th1+th2+th4])
c111=FullSimplify[(1/2)*(D[d11,th1]+D[d11,th1]-D[d11,th1])]
{{1/4 m3 (L1 y Cos[th1]+L2 y Cos[th1+th2]-x (L1 Sin[th1]+L2 Sin[th1+th2]))}}
t=d11+g1+c111
{{Ic1zz+Ic2zz+Ic3zz+g m1 (L1 Cos[th1]+1/2 L2 Cos[th1+th2])+g m2 (L1 Cos[th1]+L2
Cos[th1+th2]+1/2 L3 Cos[th1+th2+th3])+g m3 (L1 Cos[th1]+L2 Cos[th1+th2]+1/2 L4
Cos[th1+th2+th4])+m3 (1/4 (x+L1 Cos[th1]+L2 Cos[th1+th2])^2+1/4 (-y-L1 Sin[th1]-L2
Sin[th1+th2])^2)+m1 ((L1 Cos[th1]+1/2 L2 Cos[th1+th2])^2+(-L1 Sin[th1]-1/2 L2
Sin[th1+th2])^2)+1/4 m3 (L1 y Cos[th1]+L2 y Cos[th1+th2]-x (L1 Sin[th1]+L2
Sin[th1+th2]))+m2 ((L1 Cos[th1]+L2 Cos[th1+th2]+1/2 L3 Cos[th1+th2+th3])^2+(-L1
Sin[th1]-L2 Sin[th1+th2]-1/2 L3 Sin[th1+th2+th3])^2)}}
FullSimplify[d11]
{{1/4 (4 Ic1zz+4 Ic2zz+4 Ic3zz+L3^2 m2+L2^2 (m1+4 m2+m3)+L1^2 (4 (m1+m2)+m3)+m3
(x^2+y^2)+2 (L1 m3 x Cos[th1]+L1 L2 (2 m1+4 m2+m3) Cos[th2]+L2 m3 x Cos[th1+th2]+2
L2 L3 m2 Cos[th3]+2 L1 L3 m2 Cos[th2+th3]+L1 m3 y Sin[th1]+L2 m3 y Sin[th1+th2]))}}
P=m1*g*(L1*Sin[th1]+L2*Sin[th1+th2]/2)+m2*g*(L1*Sin[th1]+L2*Sin[th1+th2]+L3*Sin[t
h1+th2+th3]/2)+m3*g*((y+L1*Sin[th1]+L2*Sin[th1+th2])/2)
g m1 (L1 Sin[th1]+1/2 L2 Sin[th1+th2])+1/2 g m3 (y+L1 Sin[th1]+L2 Sin[th1+th2])+g m2
(L1 Sin[th1]+L2 Sin[th1+th2]+1/2 L3 Sin[th1+th2+th3])
g1=D[P,th1]
g m1 (L1 Cos[th1]+1/2 L2 Cos[th1+th2])+1/2 g m3 (L1 Cos[th1]+L2 Cos[th1+th2])+g m2
(L1 Cos[th1]+L2 Cos[th1+th2]+1/2 L3 Cos[th1+th2+th3])
FullSimplify[P]
1/2 g (m3 y+L1 (2 (m1+m2)+m3) Sin[th1]+L2 (m1+2 m2+m3) Sin[th1+th2]+L3 m2
Sin[th1+th2+th3])
FullSimplify[g1]
1/2 g (L1 (2 (m1+m2)+m3) Cos[th1]+L2 (m1+2 m2+m3) Cos[th1+th2]+L3 m2
Cos[th1+th2+th3])
c111=FullSimplify[(1/2)*(D[d11,th1]+D[d11,th1]-D[d11,th1])]
{{1/4 m3 (L1 y Cos[th1]+L2 y Cos[th1+th2]-x (L1 Sin[th1]+L2 Sin[th1+th2]))}}
Rc0=({
    {1, 0, 0},
```

```
    {0, 1, 0},
    {0, 0, 1}
   })
{{1,0,0},{0,1,0},{0,0,1}}
TRc0=Transpose[Rc0]
{{1,0,0},{0,1,0},{0,0,1}}
Jwc0=({
    {0},
    {0},
    {1}
   })
{{0},{0},{1}}
TJwc0=Transpose[Jwc0]
{{0,0,1}}
Ic0=({
    {Ic0xx, 0, 0},
    {0, Ic0yy, 0},
    {0, 0, Ic0zz}
   })
{{Ic0xx,0,0},{0,Ic0yy,0},{0,0,Ic0zz}}
d11=m1*TJvc1.Jvc1+m2*TJvc2.Jvc2+m3*TJvc3.Jvc3+TJwc0.Rc0.Ic0.TRc0.Jwc0+TJwc1.Rc1.Ic1.TRc1.Jwc1+TJwc2.Rc2.Ic2.TRc2.Jwc2+TJwc3.Rc3.Ic3.TRc3.Jwc3
{{Ic0zz+Ic1zz+Ic2zz+Ic3zz+m3 (1/4 (x+L1 Cos[th1]+L2 Cos[th1+th2])^2+1/4 (-y-L1 Sin[th1]-L2 Sin[th1+th2])^2)+m1 ((L1 Cos[th1]+1/2 L2 Cos[th1+th2])^2+(-L1 Sin[th1]-1/2 L2 Sin[th1+th2])^2)+m2 ((L1 Cos[th1]+L2 Cos[th1+th2]+1/2 L3 Cos[th1+th2+th3])^2+(-L1 Sin[th1]-L2 Sin[th1+th2]-1/2 L3 Sin[th1+th2+th3])^2)}}
FullSimplify[d11]
{{1/4 (4 Ic0zz+4 Ic1zz+4 Ic2zz+4 Ic3zz+L3^2 m2+L2^2 (m1+4 m2+m3)+L1^2 (4 (m1+m2)+m3)+m3 (x^2+y^2)+2 (L1 m3 x Cos[th1]+L1 L2 (2 m1+4 m2+m3) Cos[th2]+L2 m3 x Cos[th1+th2]+2 L2 L3 m2 Cos[th3]+2 L1 L3 m2 Cos[th2+th3]+L1 m3 y Sin[th1]+L2 m3 y Sin[th1+th2]))}}
c111=FullSimplify[(1/2)*(D[d11,th1]+D[d11,th1]-D[d11,th1])]
{{1/4 m3 (L1 y Cos[th1]+L2 y Cos[th1+th2]-x (L1 Sin[th1]+L2 Sin[th1+th2]))}}
```

C. Webot

```
/*

 * File:          my_controller.c

 * Date:

 * Description:

 * Author:
```

```
 * Modifications:
 */

/*
 * You may need to add include files like <webots/distance_sensor.h> or
 * <webots/differential_wheels.h>, etc.
 */
#include <webots/robot.h>
#include <webots/motor.h>
#include <webots/position_sensor.h>
#include <stdio.h>
#include <math.h>

/*
 * You may want to add macros here.
 */
#define TIME_STEP 1
#define dt 0.001
#define G   9.81
#define f   0.5
#define pi 3.141592

FILE* DATA2;
```

```
FILE* GdData;
FILE* GdData2;
FILE* GdData3;
/*
 * This is the main program.
 * The arguments of the main function can be specified by the
 * "controllerArgs" field of the Robot node
 */
 double Gd[1001] = {0};
 double Gd2[1001] = {0};
 double Gd3[1001] = {0};


int main(int argc, char **argv)
{
   GdData = fopen("IK.txt", "r");
   for(int loop = 0; loop < 1001; loop++)
   {
     fscanf(GdData, "%lf", &Gd[loop]);
     printf("%d %lf₩n",loop, Gd[loop]);
   }
   GdData2 = fopen("IK_d.txt", "r");
   for(int loop = 0; loop < 1001; loop++)
```

```
{
  fscanf(GdData2, "%lf", &Gd2[loop]);
  printf("%d %lf₩n",loop, Gd2[loop]);
}
GdData3 = fopen("IK_dd.txt", "r");
for(int loop = 0; loop < 1001; loop++)
{
  fscanf(GdData3, "%lf", &Gd3[loop]);
  printf("%d %lf₩n",loop, Gd3[loop]);
}

/* necessary to initialize webots stuff */
wb_robot_init();

WbDeviceTag m1 = wb_robot_get_device("m1");
WbDeviceTag PS1 = wb_robot_get_device("p1");
WbDeviceTag PS2 = wb_robot_get_device("p2");



int step = (int)wb_robot_get_basic_time_step();
wb_position_sensor_enable(PS1, step);
wb_position_sensor_enable(PS2, step);
```

```
wb_motor_enable_torque_feedback(m1, step);

wb_motor_set_torque(m1,0);

double theta1 = 0, preTheta1 = 0;
double theta1_dot = 0, preTheta1_dot = 0;
double theta1_ddot = 0;
double theta2 = 0;

double M_q = 0, C_q = 0, G_q = 0;
double I0 = 0.1318, I1 = 0.0275, I2 = 0.002223, I3 = 0.04172;
double L1 = 0.185, L2 = 0.438, L3 = 0.176, x = 0.5475, y=0.330;
double m = 1.7, m2 = 0.8, m3 = 2.4;

double position1 = 0;

double PD_Torque = 0;
double CTC_Torque = 0;

double Kp = 10000, Kd = 200;
double CKp = 150, Kv = 70;

double C1 =0, C2=0, C3=0, S1=0,S2=0,S12 = 0, C12=0,C23=0,C123=0;
```

```
   double t = 0, tcount = 0;

   int ttcount = 1;


   DATA2 = fopen("data2.txt", "w");

   /*

    * You should declare here WbDeviceTag variables for storing

    * robot devices like this:

    *   WbDeviceTag my_sensor = wb_robot_get_device("my_sensor");

    *   WbDeviceTag my_actuator = wb_robot_get_device("my_actuator");

    */


   /* main loop

    * Perform simulation steps of TIME_STEP milliseconds

    * and leave the loop when the simulation is over

    */

/*   while (wb_robot_step(TIME_STEP) != -1 && ttcount != 1001) {

     t = dt*tcount;


     preTheta1 = theta1;

     preTheta1_dot = theta1_dot;


     theta1 = wb_position_sensor_get_value(PS1);
```

```
        theta1_dot = (theta1-preTheta1)/dt;
        theta1_ddot = (theta1_dot-preTheta1_dot)/dt;

        double m1t = wb_motor_get_torque_feedback(m1);

        position1 = Gd[ttcount];

        wb_motor_set_position(m1,position1);

        //wb_motor_set_torque(m1,G_1);
        //wb_motor_set_torque(m2,G_2);

        //printf("%f %f₩n",Torque1 , Torque2);

        fprintf(DATA2, "%d %f %f %f %f₩n", ttcount, m1t, theta1, theta1_dot, theta1_ddot);
        printf("%d %lf₩n",ttcount, Gd[ttcount]);
        tcount++;
        ttcount++;

    };*/

/*   while (wb_robot_step(TIME_STEP) != -1) {
```

```
t = dt*tcount;

preTheta1 = theta1;
preTheta1_dot = theta1_dot;

theta1 = wb_position_sensor_get_value(PS1);
theta1_dot = (theta1-preTheta1)/dt;
theta1_ddot = (theta1_dot-preTheta1_dot)/dt;

PD_Torque = Kp*(Gd[ttcount] - theta1) - Kd*theta1_dot;

double m1t = wb_motor_get_torque_feedback(m1);

position1 = Gd[ttcount];

// wb_motor_set_position(m1,PD_Torque);

wb_motor_set_torque(m1, PD_Torque);
//wb_motor_set_torque(m2,G_2);

//printf("%f %f₩n",Torque1 , Torque2);

fprintf(DATA2, "%f %f %f %f %f₩n", position1, theta1, m1t, PD_Torque, theta1_dot);
```

```
    printf("%d %lf₩n",ttcount, Gd[ttcount]);
    tcount++;
    ttcount++;
    if(ttcount >= 1002) while(1);
};*/

while (wb_robot_step(TIME_STEP) != -1) {
    t = dt*tcount;

    preTheta1 = theta1;
    preTheta1_dot = theta1_dot;

    theta1 = wb_position_sensor_get_value(PS1);
    theta2 = wb_position_sensor_get_value(PS2);

    theta1_dot = (theta1-preTheta1)/dt;
    theta1_ddot = (theta1_dot-preTheta1_dot)/dt;

    C1 = cos(theta1);
    C2 = cos(theta2);
    C3 = cos(-0.9948374);
    S1 = sin(theta1);
    S2 = sin(theta2);
```

```
S12 = sin(theta1+theta2);

C12 = cos(theta1+theta2);

C23 = cos(theta2-0.9948374);

C123 = cos(theta1+theta2-0.9948374);


M_q = 0.25*(4*I0 + 4*I1 + 4*I2 + 4*I3 + (L3*L3)*m2 + (L2*L2)*(m+4*m2+m3) + (L1*L1)*(4*(m+m2)+m3) + m3*((x*x)+(y*y))) + 2*(L1*m3*x*C1 + L1*L2*(2*m+4*m2+m3)*C2 + L2*m3*x*C12 + 2*L2*L3*m2*C3+2*L1*L3*m2*C23 + L1*m3*y*S1 + L2*m3*y*S12);

C_q = 0.25*m3*L1*y*C1 + L2*y*C12 - x*(L1*S1 + L2*S12);

G_q = 0.5*G*(L1*(2*(m1+m2)+m3)*C1+L2*(m1+2*m2*m3)*C12+L3*m2*C123);

CTC_Torque = M_q*(CKp*(Gd[ttcount] - theta1) + Kv*(Gd2[ttcount]-theta1_dot) + Gd3[ttcount]) + C_q + G_q;


double m1t = wb_motor_get_torque_feedback(m1);


position1 = Gd[ttcount];




// wb_motor_set_position(m1,CTC_Torque);


wb_motor_set_torque(m1,CTC_Torque);

//wb_motor_set_torque(m2,G_2);


//printf("%f %f₩n",Torque1 , Torque2);
```

```
        fprintf(DATA2, "%f %f %f %f %f₩n", position1, theta1, m1t, CTC_Torque, theta1_dot);
        printf("%d %lf₩n",ttcount, M_q);
        tcount++;
        ttcount++;
        if(ttcount >= 1002) while(1);
    };

    /* Enter your cleanup code here */

    /* This is necessary to cleanup webots resources */
    wb_robot_cleanup();

    return 0;
}
```