

보행 2차 과제 보고서

휴머노이드 팀 19기 예비단원 구도연

HW_001

1. FK, IK 유도.

FK

$${}^0_1T = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^1_2T = \begin{bmatrix} \cos\theta_2 & 0 & \sin\theta_2 & 6\cos\theta_2 \\ -\sin\theta_2 & 0 & \cos\theta_2 & 6\sin\theta_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^2_3T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta_3 & -\sin\theta_3 & 0 \\ 0 & \sin\theta_3 & \cos\theta_3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

⊕ 회전 행렬

$$R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2차, 0도, 90도

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

* 1차, 0도, 90도

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

$R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

$$\Rightarrow {}^0_3T = {}^0_1T \times {}^1_2T \times {}^2_3T$$

$$\Rightarrow x = 6\cos\theta_1(\cos\theta_2 + \cos\theta_3) + 6\cos\theta_1\cos(\theta_2+\theta_3)$$

$$y = 6\sin\theta_1(\cos\theta_2 + \cos\theta_3) + 6\sin\theta_1\cos(\theta_2+\theta_3)$$

$$z = 6 + 6\sin\theta_2 + 6\sin(\theta_2+\theta_3)$$

* DH 변환 행렬을 구함.

$$T_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \cos d_i & \sin\theta_i \cos d_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\theta_i \cos d_i & -\sin\theta_i \cos d_i & a_i \sin\theta_i \\ 0 & \sin d_i & \cos d_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_1 = \begin{bmatrix} \cos\theta_1 & 0 & \sin\theta_1 & 0 \\ \sin\theta_1 & 0 & -\cos\theta_1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_2 = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & 6\cos\theta_2 \\ \sin\theta_2 & \cos\theta_2 & 0 & 6\sin\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_3 = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & 6\cos\theta_3 \\ \sin\theta_3 & \cos\theta_3 & 0 & 6\sin\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$x = 6\cos\theta_1(\cos\theta_2 + \cos\theta_3) + 6\cos\theta_1\cos(\theta_2+\theta_3)$$

$$y = 6\sin\theta_1(\cos\theta_2 + \cos\theta_3) + 6\sin\theta_1\cos(\theta_2+\theta_3)$$

$$z = 6 + 6\sin\theta_2 + 6\sin(\theta_2+\theta_3)$$

IK

$$\theta_1 = \arctan2(x, y)$$

$$r = \sqrt{x^2 + y^2}$$

$$z = l_1 - z$$

$$p = \frac{r^2 + z^2 - l_2^2 - l_3^2}{2 \cdot l_2 \cdot l_3} \quad (0 \leq -1, 1 \text{ 안함})$$

$$\theta_2 = \arccos(p)$$

$$\theta_3 = \arctan2(z, r) - \arctan2(l_2 \sin(\theta_2), l_2 \cos(\theta_2))$$

2. 매틀랩 코드

<FK>

```
clc;
clear;
close all;

% 링크 길이 정의
L1 = 4; % 첫 번째 링크 길이
L2 = 9; % 두 번째 링크 길이
L3 = 8; % 세 번째 링크 길이

% 각도 입력
disp('각 링크의 각도를 입력하세요. ');
theta1 = input('Theta1 (z축 회전): ') * pi / 180; % 라디안 변환
theta2 = input('Theta2 (y축 회전): ') * pi / 180;
theta3 = input('Theta3 (x축 회전): ') * pi / 180;

% DH 파라미터를 사용한 변환 행렬 정의 함수
DH_transform = @(theta, d, a, alpha) ...
    [cos(theta), -sin(theta)*cos(alpha), sin(theta)*sin(alpha), a*cos(theta);
     sin(theta), cos(theta)*cos(alpha), -cos(theta)*sin(alpha), a*sin(theta);
     0,          sin(alpha),          cos(alpha),          d;
     0,          0,          0,          1];

% 각 링크의 변환 행렬 계산
% T_0_1: 링크 1 변환 (z축 회전 및 z축 이동)
T_0_1 = DH_transform(theta1, L1, 0, pi/2);

% T_1_2: 링크 2 변환 (y축 회전 및 x축 이동)
T_1_2 = DH_transform(theta2, 0, L2, 0);

% T_2_3: 링크 3 변환 (x축 회전 및 x축 이동)
T_2_3 = DH_transform(theta3, 0, L3, 0);

% 총 변환 행렬 계산
T_0_3 = T_0_1 * T_1_2 * T_2_3;

% 조인트 위치 계산
origin = [0; 0; 0; 1]; % 기준점 (원점)
```

```

p1 = T_0_1 * origin;    % 링크 1 끝점 (어깨)
p2 = T_0_1 * T_1_2 * origin; % 링크 2 끝점 (팔꿈치)
p3 = T_0_3 * origin;    % 링크 3 끝점 (엔드 이펙터)

% 좌표 추출
jointsX = [origin(1), p1(1), p2(1), p3(1)];
jointsY = [origin(2), p1(2), p2(2), p3(2)];
jointsZ = [origin(3), p1(3), p2(3), p3(3)];

% 좌표 출력
disp('조인트 위치 좌표:');
disp(['어깨: (', num2str(p1(1)), ', ', num2str(p1(2)), ', ', num2str(p1(3)), ')']);
disp(['팔꿈치: (', num2str(p2(1)), ', ', num2str(p2(2)), ', ', num2str(p2(3)), ')']);
disp(['엔드 이펙터: (', num2str(p3(1)), ', ', num2str(p3(2)), ', ', num2str(p3(3)), ')']);

% 3D 그래프 시각화
figure;
hold on;
grid on;
axis equal;

% 링크별 시각화 (각 링크는 서로 다른 색상)
plot3(jointsX, jointsY, jointsZ, 'k--', 'LineWidth', 1, 'DisplayName', 'Robot Path');
% 전체 경로
plot3([jointsX(1), jointsX(2)], [jointsY(1), jointsY(2)], [jointsZ(1), jointsZ(2)], ...
      'r', 'LineWidth', 3, 'DisplayName', 'Link 1');
plot3([jointsX(2), jointsX(3)], [jointsY(2), jointsY(3)], [jointsZ(2), jointsZ(3)], ...
      'g', 'LineWidth', 3, 'DisplayName', 'Link 2');
plot3([jointsX(3), jointsX(4)], [jointsY(3), jointsY(4)], [jointsZ(3), jointsZ(4)], ...
      'b', 'LineWidth', 3, 'DisplayName', 'Link 3');

% 엔드 이펙터 위치 강조
scatter3(jointsX(4), jointsY(4), jointsZ(4), 100, 'filled', 'MarkerFaceColor', 'k', ...
        'DisplayName', 'End Effector');

% 그래프 설정
xlabel('X-axis');
ylabel('Y-axis');
zlabel('Z-axis');

```

```

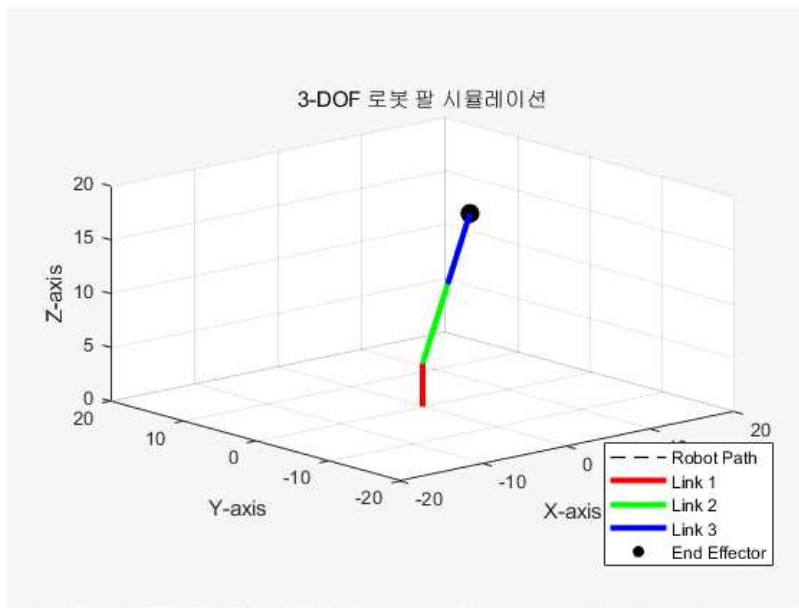
title('3-DOF 로봇 팔 시뮬레이션');
legend;
view(3);

% 축 범위 설정
xlim([-20, 20]);
ylim([-20, 20]);
zlim([0, 20]);

```

<실행 결과>

각도 입력, (30, 40, 0)



<IK>

```
clc;
clear;
close all;

% 링크 길이 정의
L1 = 4; % 첫 번째 링크 길이
L2 = 9; % 두 번째 링크 길이
L3 = 8; % 세 번째 링크 길이

% 엔드 이펙터 목표 위치 입력
disp('엔드 이펙터의 목표 위치를 입력하세요:');
x = input('X 좌표: ');
y = input('Y 좌표: ');
z = input('Z 좌표: ');

% IK 계산
% Step 1: \theta_1 계산
theta1 = atan2(y, x);

% Step 2: z'와 r 계산
z_prime = z - L1; % 수직 높이 차이
r = sqrt(x^2 + y^2); % x-y 평면 거리

% Step 3: D 계산 (코사인 법칙)
D = (r^2 + z_prime^2 - L2^2 - L3^2) / (2 * L2 * L3);

% 작업 공간 초과 확인
if abs(D) > 1
    error('목표 위치가 로봇 팔의 작업 범위를 초과했습니다.');
```

end

```
% Step 4: \theta_3 계산
theta3 = acos(D);

% Step 5: \theta_2 계산
theta2 = atan2(z_prime, r) - atan2(L3 * sin(theta3), L2 + L3 * cos(theta3));

% 결과 출력
disp('계산된 관절 각도 (라디안):');
```

```

disp(['Theta1: ', num2str(theta1)]);
disp(['Theta2: ', num2str(theta2)]);
disp(['Theta3: ', num2str(theta3)]);

disp('계산된 관절 각도 (도):');
disp(['Theta1: ', num2str(rad2deg(theta1)), '°']);
disp(['Theta2: ', num2str(rad2deg(theta2)), '°']);
disp(['Theta3: ', num2str(rad2deg(theta3)), '°']);

% FK 검증
% 변환 행렬 정의 함수
RotZ = @(theta) [cos(theta), -sin(theta), 0, 0;
                 sin(theta),  cos(theta), 0, 0;
                 0,          0,          1, 0;
                 0,          0,          0, 1];
RotX = @(theta) [1, 0,          0,          0;
                 0, cos(theta), -sin(theta), 0;
                 0, sin(theta),  cos(theta), 0;
                 0, 0,          0,          1];
Trans = @(x, y, z) [1, 0, 0, x;
                   0, 1, 0, y;
                   0, 0, 1, z;
                   0, 0, 0, 1];

% FK로 계산된 엔드 이펙터 위치
T_0_1 = RotZ(theta1) * Trans(0, 0, L1);
T_1_2 = RotX(theta2) * Trans(0, 0, L2);
T_2_3 = RotX(pi / 2) * RotZ(theta3) * Trans(L3, 0, 0);
T_0_3 = T_0_1 * T_1_2 * T_2_3;

% FK 계산 결과 추출
calculated_x = T_0_3(1, 4);
calculated_y = T_0_3(2, 4);
calculated_z = T_0_3(3, 4);

disp('FK로 계산된 엔드 이펙터 위치:');
disp(['X: ', num2str(calculated_x)]);
disp(['Y: ', num2str(calculated_y)]);
disp(['Z: ', num2str(calculated_z)]);

```

```

% 3D 그래프 시각화
% 각 링크의 조인트 좌표 계산
origin = [0; 0; 0; 1];
p1 = T_0_1 * origin;
p2 = T_0_1 * T_1_2 * origin;
p3 = T_0_3 * origin;

jointsX = [origin(1), p1(1), p2(1), p3(1)];
jointsY = [origin(2), p1(2), p2(2), p3(2)];
jointsZ = [origin(3), p1(3), p2(3), p3(3)];

figure;
hold on;
grid on;
axis equal;

% 링크 시각화
plot3(jointsX, jointsY, jointsZ, 'k--', 'LineWidth', 1, 'DisplayName', 'Robot Path');
plot3([jointsX(1), jointsX(2)], [jointsY(1), jointsY(2)], [jointsZ(1), jointsZ(2)], ...
      'r', 'LineWidth', 3, 'DisplayName', 'Link 1');
plot3([jointsX(2), jointsX(3)], [jointsY(2), jointsY(3)], [jointsZ(2), jointsZ(3)], ...
      'g', 'LineWidth', 3, 'DisplayName', 'Link 2');
plot3([jointsX(3), jointsX(4)], [jointsY(3), jointsY(4)], [jointsZ(3), jointsZ(4)], ...
      'b', 'LineWidth', 3, 'DisplayName', 'Link 3');

% 엔드 이펙터 강조
scatter3(jointsX(4), jointsY(4), jointsZ(4), 100, 'filled', 'MarkerFaceColor', 'k', ...
        'DisplayName', 'End Effector');

% 그래프 설정
xlabel('X-axis');
ylabel('Y-axis');
zlabel('Z-axis');
title('3-DOF 로봇 팔 IK 계산 및 FK 검증');
legend;
view(3);

% 축 범위 설정
xlim([-20, 20]);
ylim([-20, 20]);

```

```
zlim([0, 20]);
```

<실행 결과>

입력 값: (10, 7, 0)

