

보행_DAY3

2축을 통해 '☆', '□', '△', '○'순으로 목표 위치에 도달하는 것을 보여주는 코드이다.

```
% 2관절 RR 매니퓰레이터의 Inverse Kinematics(IK) 및 목표 위치 이동 시뮬레이션
clear; clc;
close all;

% 매니퓰레이터 링크 길이 설정
L1 = 8; % 첫 번째 링크 길이
L2 = 8; % 두 번째 링크 길이

% 목표 위치 및 마커 설정
targets = [5, 5; 5, 10; 10, 5; 10, 10];
markers = {'☆', '□', '△', '○'};

% 그래프 설정
figure;
hold on;
xlabel('X');
ylabel('Y');
title('2 DOF');
grid on;
axis equal;

% 목표 위치 마커 표시
for i = 1:size(targets, 1)
    text(targets(i,1), targets(i,2), markers{i}, ...
        'FontSize', 14, 'HorizontalAlignment', 'center');
end

% 각 목표 위치로 이동
current_theta1 = 0;
current_theta2 = 0;

for i = 1:size(targets, 1)
    % 현재 목표 위치
    x_target = targets(i, 1);
    y_target = targets(i, 2);

    % Inverse Kinematics 계산
    r = sqrt(x_target^2 + y_target^2);
    phi = atan2(y_target, x_target);
    cos_theta2 = (r^2 - L1^2 - L2^2) / (2 * L1 * L2);
```

```

% 목표 위치 도달 가능 여부 확인
if abs(cos_theta2) > 1
    warning('목표 지점 (%.2f, %.2f)에 도달할 수 없습니다.', x_target, y_target);
    continue;
end

% theta1, theta2 계산 (elbow-up 형태)
sin_theta2 = sqrt(1 - cos_theta2^2);
theta2_target = atan2(sin_theta2, cos_theta2);
theta1_target = phi - atan2(L2 * sin_theta2, L1 + L2 * cos_theta2);

% 현재 위치에서 목표 위치까지 보간
steps = linspace(0, 1, 50);

for t = steps
    % 각도 보간
    theta1 = (1-t) * current_theta1 + t * theta1_target;
    theta2 = (1-t) * current_theta2 + t * theta2_target;

    % 현재 말단장치 위치 계산
    x_mid = L1 * cos(theta1);
    y_mid = L1 * sin(theta1);
    x_end = x_mid + L2 * cos(theta1 + theta2);
    y_end = y_mid + L2 * sin(theta1 + theta2);

    % 매니플레이터 시각화
    cla;
    plot([0, x_mid, x_end], [0, y_mid, y_end], '-o', 'LineWidth', 1.5);

    % 목표 위치 마커 다시 표시
    for j = 1:size(targets, 1)
        text(targets(j,1), targets(j,2), markers{j}, ...
            'FontSize', 14, 'HorizontalAlignment', 'center');
    end

    % 그래프 설정 유지
    grid on;
    axis equal;
    axis([-10 20 -10 20]);

    % 애니메이션 효과를 위한 일시 정지
    pause(0.05);
end

% 현재 각도 업데이트
current_theta1 = theta1_target;
current_theta2 = theta2_target;
end

```

hold off;