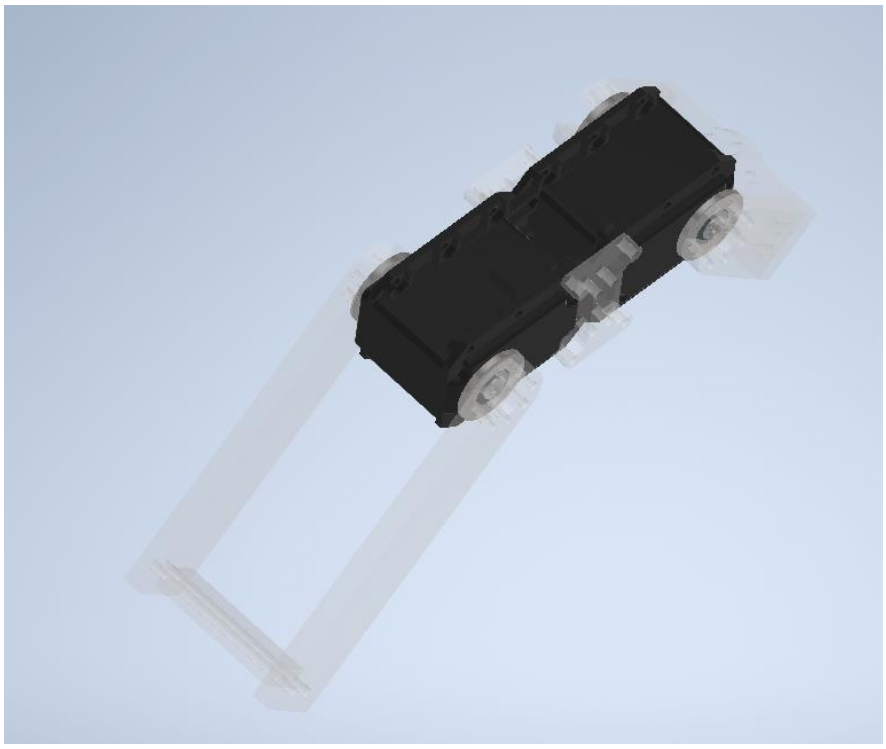
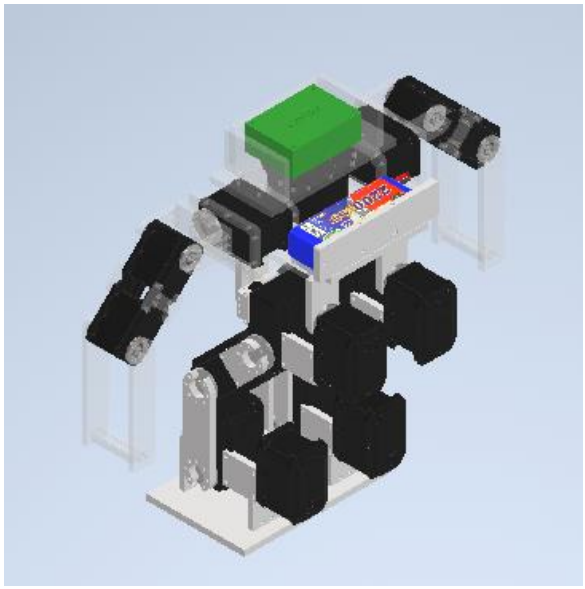
격투로봇팔 기구학 해석

19기 예비단원 이원준

1. 격투 로봇 팔 구성





2. 순기구학 해석 코드

```
clc;
clear;
close all;

%% 기본 설정
% 링크 길이 (mm 단위)
link_lengths = [36, 77.2, 100]; % 링크 1, 2, 3의 길이

% 초기 조인트 각도 (라디안 단위)
theta1 = deg2rad(0); % 조인트 1 초기값
theta2 = deg2rad(0); % 조인트 2 초기값
theta3 = deg2rad(0); % 조인트 3 초기값

% 조인트 개수
n = 3; % 총 3개의 조인트

% 동기구학 계산 함수
T_single = @(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;
];

%% 매니퓰레이터 초기 설정
figure; % 그래프를 출력할 새 창 생성
updateManipulator(theta1, theta2, theta3);

disp('매니퓰레이터 초기 상태가 설정되었습니다.');
disp('조인트 각도를 입력하여 매니퓰레이터를 업데이트하세요.');
disp('예: theta1 = 30; theta2 = 45; theta3 = 60;');

%% 사용자 입력 대기 및 업데이트 루프
while true
    % 사용자 입력 받기
    user_input = input('각도를 입력하세요 (exit 입력 시 종료): ', 's');

    % 종료 조건
    if strcmpi(user_input, 'exit')
        disp('프로그램을 종료합니다.');
        break;
    end

    % 사용자 입력 실행
    try
        % 사용자 입력 실행 및 업데이트
        eval(user_input); % 입력된 명령 실행
        theta1 = deg2rad(theta1); % 각도를 라디안으로 변환
        theta2 = deg2rad(theta2);
        theta3 = deg2rad(theta3);

        % 매니퓰레이터 업데이트
        updateManipulator(theta1, theta2, theta3);
    catch
        disp('잘못된 입력입니다. 다시 입력하세요.');
    end
end

%% 매니퓰레이터 업데이트 함수
function updateManipulator(theta1, theta2, theta3)
    % 링크 길이
    link_lengths = [36, 77.2, 100];

    % DH 테이블 업데이트
    DH_table = [
        theta1, link_lengths(1), 0,      pi/2;
        theta2, 0,               link_lengths(2), 0;
        theta3, 0,               link_lengths(3), 0;
    ];

    % 동기구학 계산
    T = eye(4); % 초기 변환 행렬
    positions = [0, 0, 0]; % 기준 프레임의 원점

    for i = 1:size(DH_table, 1)
        theta = DH_table(i, 1);
        d = DH_table(i, 2);
        a = DH_table(i, 3);
        alpha = DH_table(i, 4);

        % 현재 조인트의 변환 행렬 계산
        T_i = T_single(theta, d, a, alpha);
        T = T * T_i; % 누적 변환 행렬
        positions = [positions; T(1:3, 4)']; % 각 조인트 위치 저장
    end

    % 매니퓰레이터 그래프 플로팅
    plot3(positions(:, 1), positions(:, 2), positions(:, 3), '-o', 'LineWidth', 2);
    grid on;
    xlabel('X (mm)');
    ylabel('Y (mm)');
    zlabel('Z (mm)');
    title('Manipulator Forward Kinematics');
    axis equal;
    view(3); % 3D 뷰 설정
    drawnow; % 그래프 업데이트
end
```

```
clc;
clear;
close all;
%% 기본 설정
% 링크 길이 (mm 단위)
link_lengths = [36, 77.2, 100]; % 링크 1, 2, 3의 길이
% 초기 조인트 각도 (라디안 단위)
theta1 = deg2rad(0); % 조인트 1 초기값
theta2 = deg2rad(0); % 조인트 2 초기값
theta3 = deg2rad(0); % 조인트 3 초기값
% 조인트 개수
n = 3; % 총 3개의 조인트
%% 매니퓰레이터 초기 생성
figure; % 그래프를 출력할 새 창 생성
updateManipulator(theta1, theta2, theta3);
disp('매니퓰레이터 초기 상태가 생성되었습니다.');
disp('조인트 각도를 입력하여 매니퓰레이터를 업데이트하세요.');
disp('예: theta1 = 30; theta2 = 45; theta3 = 60;');
%% 사용자 입력 대기 및 업데이트 루프
while true
% 사용자 입력 받기
user_input = input('각도를 입력하세요 (exit 입력 시 종료): ', 's');
% 종료 조건
if strcmpi(user_input, 'exit')
disp('프로그램을 종료합니다.');
break;
end
% 사용자 입력 실행
try
% 사용자 입력 실행 및 업데이트
eval(user_input); % 입력된 명령 실행
theta1 = deg2rad(theta1); % 각도를 라디안으로 변환
theta2 = deg2rad(theta2);
theta3 = deg2rad(theta3);
% 매니퓰레이터 업데이트
updateManipulator(theta1, theta2, theta3);
catch
disp('잘못된 입력입니다. 다시 입력하세요.');
end
end
%% 매니퓰레이터 업데이트 함수
function updateManipulator(theta1, theta2, theta3)
% 링크 길이
link_lengths = [36, 77.2, 100];
% 순기구학 계산 함수 정의 (지역 함수)
T_single = @(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;
];
% DH 테이블 업데이트
DH_table = [
theta1, link_lengths(1), 0, pi/2;
theta2, 0, link_lengths(2), 0;
theta3, 0, link_lengths(3), 0;
```

```
];
% 순기구학 계산
T = eye(4); % 초기 변환 행렬
positions = [0, 0, 0]; % 기준 프레임의 원점
for i = 1:size(DH_table, 1)
theta = DH_table(i, 1);
d = DH_table(i, 2);
a = DH_table(i, 3);
alpha = DH_table(i, 4);
% 현재 조인트의 변환 행렬 계산
T_i = T_single(theta, d, a, alpha);
T = T * T_i; % 누적 변환 행렬
positions = [positions; T(1:3, 4)']; % 각 조인트 위치 저장
end
% 매니퓰레이터 그래프 플로팅
plot3(positions(:, 1), positions(:, 2), positions(:, 3), '-o', 'LineWidth', 2);
grid on;
xlabel('X (mm)');
ylabel('Y (mm)');
zlabel('Z (mm)');
title('Manipulator Forward Kinematics');
axis equal;
view(3); % 3D 뷰 설정
drawnow; % 그래프 업데이트
end
```

2-1. 순기구학 코드 설명

```
while true
    user_input = input('각도를 입력하세요 (exit 입력 시 종료): ', 's');
    if strcmpi(user_input, 'exit')
        disp('프로그램을 종료합니다.');
        break;
    end
    try
        eval(user_input);
        theta1 = deg2rad(theta1);
        theta2 = deg2rad(theta2);
        theta3 = deg2rad(theta3);
        updateManipulator(theta1, theta2, theta3);
    catch
        disp('잘못된 입력입니다. 다시 입력하세요.');
    end
end
```

- 사용자에게 회전 각도를 입력받고 매니퓰레이터를 그래프로 표현함

```
DH_table = [
    theta1, link_lengths(1), 0,          pi/2;
    theta2, 0,                    link_lengths(2), 0;
    theta3, 0,                    link_lengths(3), 0;
];
```

- 입력받은 값을 DH 테이블로 표현함

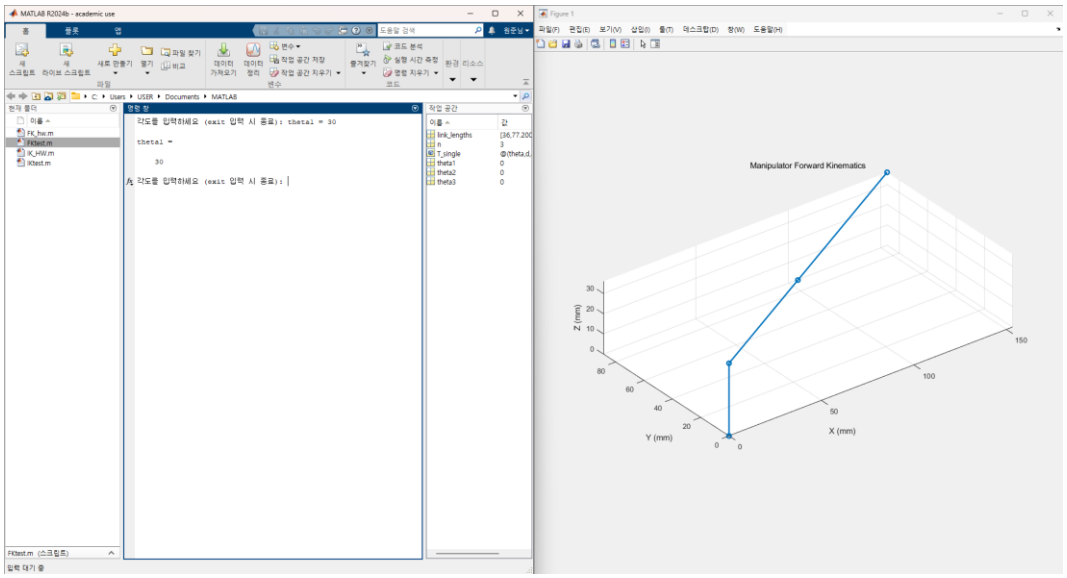
```
T = eye(4);
positions = [0, 0, 0];
```

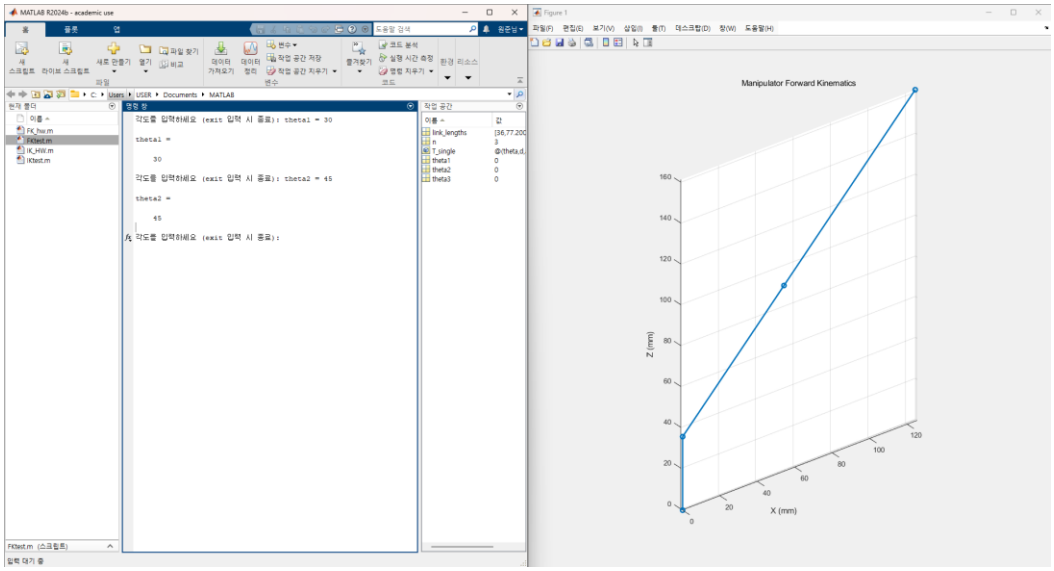
- T는 기준 좌표계에서의 변환행렬을 의미한다.
- Positions는 각 조인트의 위치를 저장할 행렬을 의미한다.

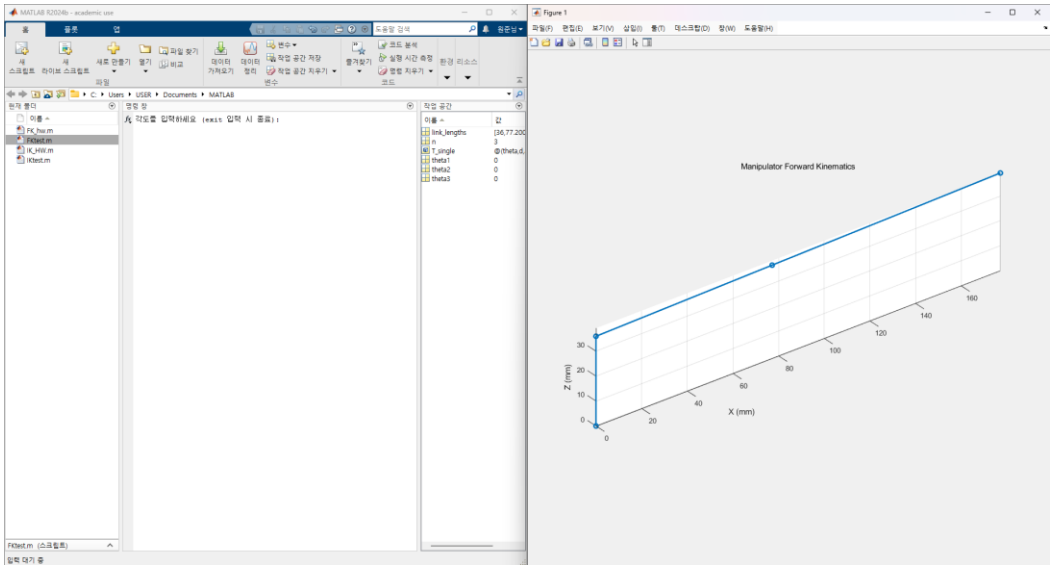
```
for i = 1:size(DH_table, 1)
    theta = DH_table(i, 1);
    d = DH_table(i, 2);
    a = DH_table(i, 3);
    alpha = DH_table(i, 4);

    T_i = T_single(theta, d, a, alpha);
    T = T * T_i;
    positions = [positions; T(1:3, 4)'];
end
```

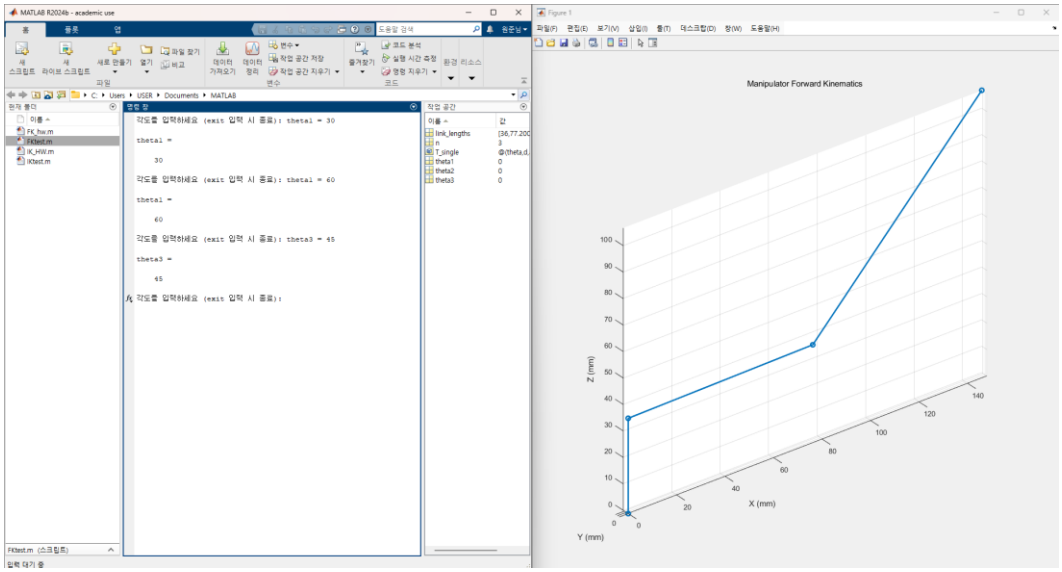
- 각 조인트의 변환 행렬(`T_i`)을 계산하고 누적하여 전체 매니퓰레이터의 위치를 계산한다.
- 이때 매 조인트의 위치를 positions에 좌표를 저장한다.

2-2. 실행 결과


각도를 입력하세요 (exit 입력 시 종료): theta1 = 30
theta1 =
30
각도를 입력하세요 (exit 입력 시 종료):
Manipulator Forward Kinematics
Z (mm)
Y (mm)
X (mm)


MATLAB R2024b - academic use
Figure 1
Manipulator Forward Kinematics
Z (mm)
X (mm)
명령 창
작업 공간
각도를 입력하세요 (exit 입력 시 종료): theta1 = 30
theta1 =
30
각도를 입력하세요 (exit 입력 시 종료): theta2 = 45
theta2 =
45
각도를 입력하세요 (exit 입력 시 종료):


MATLAB R2024b - academic use
Figure 1
Manipulator Forward Kinematics
Z (mm)
X (mm)
명령 창
작업 공간
각도를 입력하세요 (exit 입력 시 종료):


MATLAB R2024b - academic use
Figure 1
Manipulator Forward Kinematics
Z (mm)
X (mm)
Y (mm)