

EEE4320-01

Digital Control Engineering

HW#2

2018142059

김서영

1. MATLAB code

%% constants, simulation time, motor commands, calculation

매트랩 코드의 앞부분에서는 주어진 상수인 로봇 몸체의 지름과 로봇 바퀴사이의 거리, 총 시뮬레이션 시간과 time step의 시간을 정하고, simulation에 사용되는 kinematics 공식들을 정리하였다.

%% robot simulation

Simulation 섹션에서는 initial position과 initial head angle을 초기화하고, 실시간으로 그래프를 그리기 위한 array들을 지정하였다. Simulation loop 부분에서 로봇의 움직임을 정의하는데, 먼저 입력된 command period에 따라 random motor command가 양쪽 바퀴의 각속도로 주어진다. 그러면 이에 따라 각 바퀴의 속력을 구한다. 이 때 두 바퀴의 속력에 따라 로봇의 경로가 변화하는데, 이는 다음과 같은 세 가지 경우로 나눌 수 있다.

V_L 이 V_R 보다 작을 때, 로봇은 우회전한다.

$V_L = V_R$ 일 때, 로봇은 직진한다.

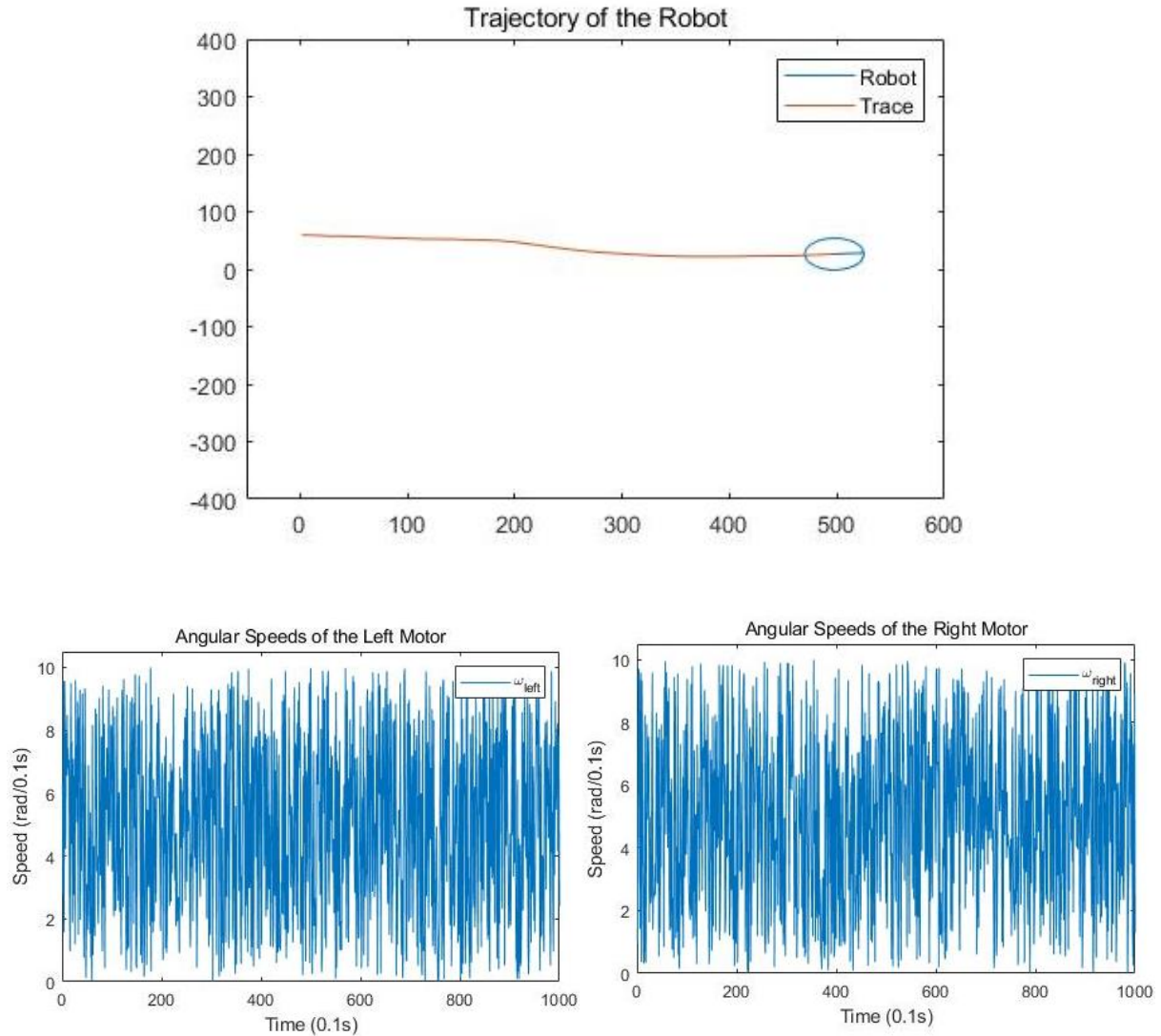
V_L 이 V_R 보다 클 때, 로봇은 좌회전한다.

따라서 코드에서도 각각의 경우를 if문으로 따져 로봇의 움직임을 서술한다. 로봇의 양 바퀴의 속력이 달라 회전할 때는 회전 반지름 R 과 각속도 w 를 먼저 구하고, 이에 따른 로봇의 위치와 head angle의 변화를 서술한다. 로봇이 직진할 때는 R 과 w 가 없으므로 head angle은 그대로이고, 로봇의 위치 변화만을 서술하였다.

그 후 plot을 위해 앞에서 정의한 array에 값을 넣고, 로봇과 로봇의 경로, 각 바퀴의 각속도를 실시간으로 plot하도록 코드를 작성하였다.

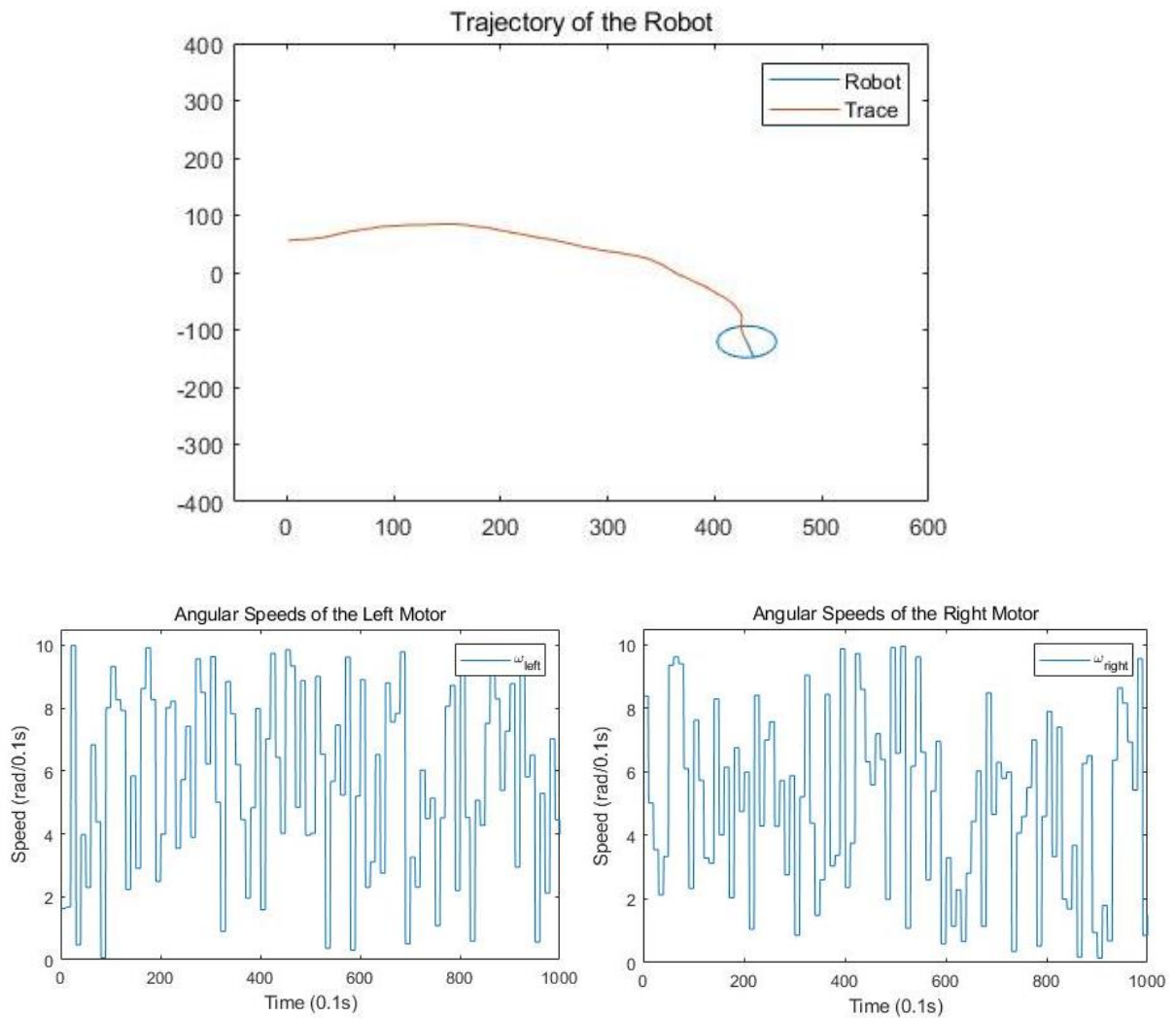
2. Simulation Results (바퀴 반지름=1mm)

(1) 로봇이 매 time step마다 무작위로 움직일 때 (command_period = 1)



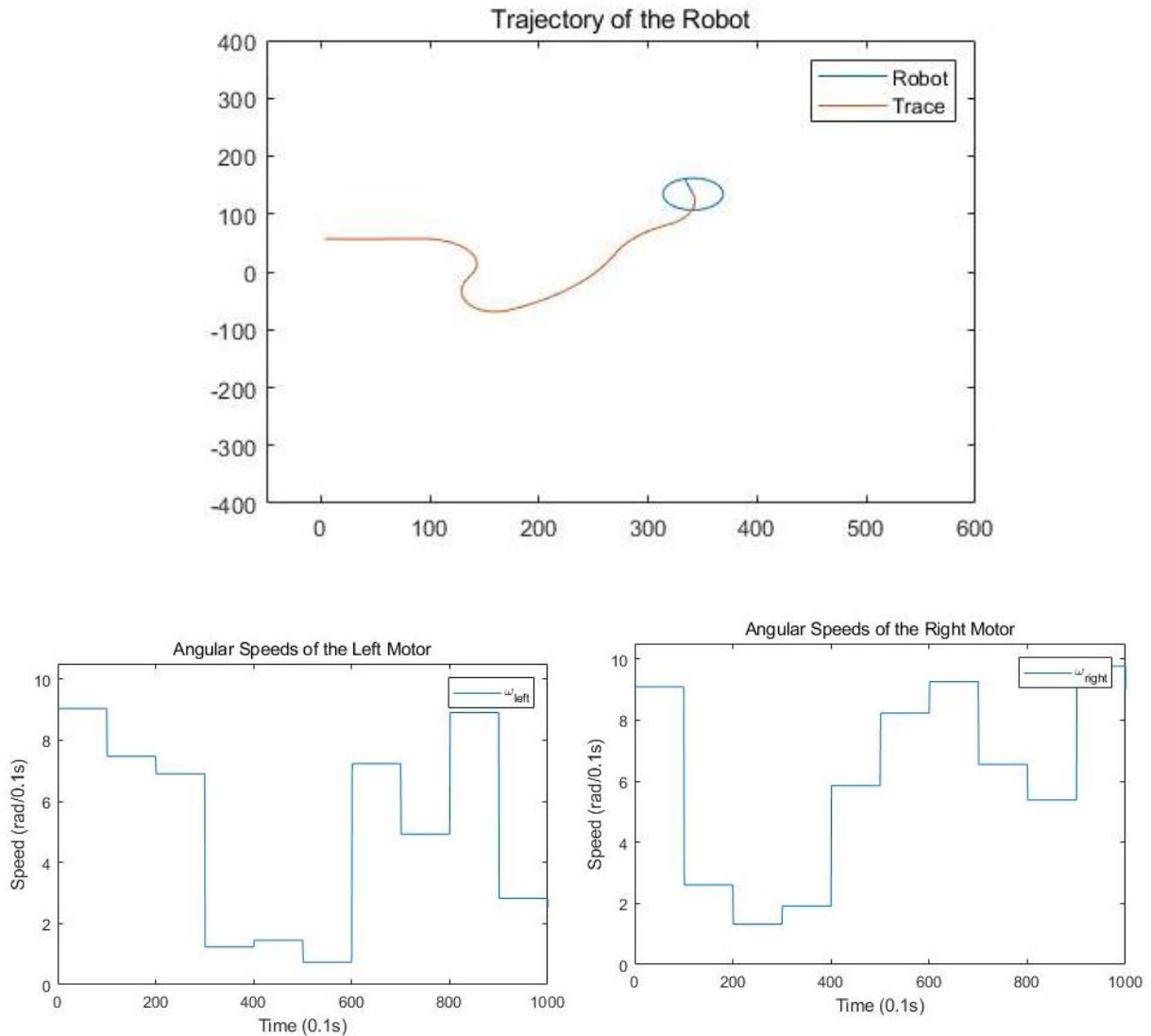
Random motor command가 매 time step=dt=0.1초마다 내려지기 때문에 전환된 방향으로의 주행이 경로에서 눈에 띄지 않을 만큼 지속되지 않는다. 따라서 경로에서 눈에 띄는 방향전환 없이 +x방향으로 나아가는 형태로 나타난다.

(2) 로봇이 10 time step마다 내려지는 무작위 motor command를 따를 때
(command_period = 10)



Random motor command가 10 time step=10dt=1초마다 내려질 때, (1)의 경우보다 더 적은 command의 변화로 곡선 구간이 더 길게 나타나 전체적인 경로 또한 (1)보다는 경로에서의 방향 전환이 더 잘 보이는 형태로 나타난다.

- (3) 로봇이 100 time step마다 내려지는 무작위 motor command를 따를 때
(command_period = 100)



Random motor command가 100 time step=100dt=10초마다 내려지는 경우로, 방향 전환을 한 후 충분한 이동시간이 주어지기 때문에 앞의 두 경우보다 곡선 구간이 더 확연하게 나타난다.

또한, command period가 길어서 모터의 각속도 차이에 따른 방향 전환을 확인할 수 있는데, 실제 kinematics에 따라 앞에서 코딩한 대로 왼쪽 바퀴의 모터의 각속도가 더 빠르면 우측으로 회전하고, 오른쪽 바퀴의 모터의 각속도가 더 빠르면 왼쪽으로 회전하는 것을 확인할 수 있었다.

3. Source Code

Matlab code
<pre>%% Digital Control Engineering & Robotics HW#2 % 2018142059 김서영 %% constants L_r=55; %[mm] robot diameter L_b=50; %[mm] length of wheel base wr=1; %[mm] radius of robot wheel %% simulation time total_time=100; %[s] Total simulation time dt = 0.1; %[s] time step %% motor commands W_l = rand*10; %left angular speed W_r = rand*10; %right angular speed %% calculation V_l = wr*W_l; %left speed V_r = wr*W_r; %right speed V = (V_l+V_r)/2; %average speed R = (L_b*(V_l+V_r))/(2*(V_r-V_l)); %moving circle radius w = (V_r - V_l)/L_b; %moving angular speed %% robot simulation %simulation initiate x = rand * 10; y = rand * 10 + 50; %initial position head = 0; %initial head angle hold off; %for plot trace_x = zeros(1, total_time/dt); %x axis trace trace_y = zeros(1, total_time/dt); %y axis trace aspeed_l = zeros(1, total_time/dt); %left angular speed aspeed_r = zeros(1, total_time/dt); %right angular speed n=0; %time step %simulation for loop for i = 0:dt:total_time %simulation n = n+1; %nth time step %commands command_period = 1; %command period if (mod(i,command_period*dt)==0) %random exploration motor command</pre>

```

        W_l = rand*10;
        W_r = rand*10;
    end

    %speed
    V_l = wr*W_l;           %left speed
    V_r = wr*W_r;           %right speed
    V = (V_l+V_r)/2;        %average speed

    %movement
    if(V_l < V_r)            %turning left
        R = (L_b*(V_l+V_r))/(2*(V_r-V_l));
        w = (V_r - V_l)/L_b;

        x = x + R * sin(head+w*dt) - R * sin(head);
        y = y - R * cos(head+w*dt) + R * cos(head);
        head = head + w*dt;

    elseif(V_l>V_r)         %turning right
        R = (L_b*(V_l+V_r))/(2*(V_l-V_r));
        w = (V_r - V_l)/L_b;

        x = x - R * sin(head+w*dt) + R * sin(head);
        y = y + R * cos(head+w*dt) - R * cos(head);
        head = head + w*dt;

    else                    %go straight
        x = x + cos(head)*V*dt;
        y = y + sin(head)*V*dt;
        head=head;
    end

    %value for plot
    trace_x(n) = x;
    trace_y(n) = y;
    aspeed_l(n) = W_l;
    aspeed_r(n) = W_r;

    %plot
    %robot drawing
    subplot(3,1,1);
    t = 0:0.1:2*pi+0.2;
    r = 55/2;                % position change
    rx = x + L_r/2 * cos(t); ry = y + L_r/2 * sin(t);
    p1 = plot(rx,ry,'DisplayName','Robot');
    %draw robot body
    hold on;
    p2 = line([x x+L_r/2*cos(head)], [y
y+L_r/2*sin(head)]);        % draw robot head

```

```

    axis([-100 500 -400 400]); % axis
    p3 = plot(trace_x(1:n),
trace_y(1:n), 'DisplayName', 'Trace'); % draw robot trace
    title('Trajectory of the Robot')
    legend([p1,p3]);
    hold off;

    %angular speed plot
    subplot(3,1,2);
    s1 =
plot(aspeed_l(1:n), 'DisplayName', '\omega_{left}');
%left angular speed
    title('Angular Speeds of the Left Motor');
    ylabel('Speed (cm/s)'); xlabel('Time (0.1s)');
    legend();
    axis([0 total_time/dt 0 10.5]);

    subplot(3,1,3);
    s2 =
plot(aspeed_r(1:n), 'DisplayName', '\omega_{right}');
%right angular speed
    title('Angular Speeds of the Right Motor');
    ylabel('Speed (cm/s)'); xlabel('Time (0.1s)');
    legend();
    axis([0 total_time/dt 0 10.5]);

    pause(0.1);
end

```