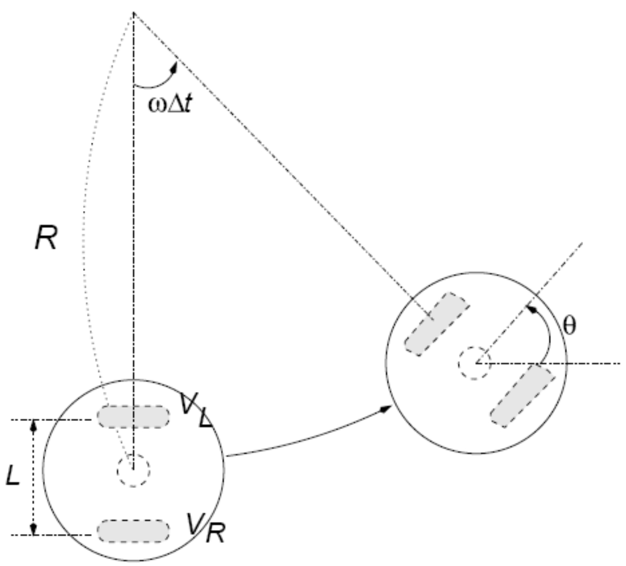
2021-1 Control Engineering

Due date: 14 April, 2021   (before the beginning of class)

We will test robotic behaviours with wheel motors. A robot is moving around with its two wheels. The robot can control each of the two wheel motors with an appropriate speed.

If the left motor speed is greater than the right motor speed, then the robot will draw a circular movement clockwise. If the right motor speed is greater, the robot will have a circular movement anticlockwise. If both speeds are the same, the robot will move straightforward.



You can simulate the dynamic equation of wheel motor system as below:

$$I\, dw/dt + c\, w = T$$

where I is the moment of inertia, c is the damping constant, w is the angular speed of the wheel motor, and T is the torque given to the motor system. The dynamic equation is changed into a discrete equation.

$$I\, [w(n)-w(n-1)] \,/\, \Delta t + c\, w(n-1) = T(n)$$

Thus, we can write it with the following recursive equation:

$$W(n) = w(n-1) + \Delta t \,/\, I * (-c*w(n-1) + T(n))$$

The recursive equation can be simulated with the MATLAB program written below:

```
% consider the first-order system  I dw/dt + c w = T
% w : angular speed of a wheel, T: torque, I: inertia moment,
% c: damping constant

I=1; c=1; dt=0.1;
t = ones(1000,1);  % torque is given with a step function
t(1)=0;

w(1)=0;  w2(1)=0;  % initialization
for i=2:1000, w(i)=w(i-1) + dt/I*(-c*w(i-1) + t(i)); end;
plot((1:1000)*dt-dt, w) % draw the angular speed with step torque
axis([0 5 0 1]); hold on;

dt = 0.01;  % you can see the effect of time step
for i=2:1000, w2(i)=w2(i-1) + dt/I*(-c*w2(i-1) + t(i)); end;
plot((1:1000)*dt-dt, w2) % draw the angular speed
axis([0 5 0 1]); hold on;

% continuous system
sys = tf(1, [I, c])  % step function for 1 / (Is+ c)
step(sys)  % step function for 1 / (Is+ c)

% you can compare the continuous system with discrete equations
legend(["dt = 0.1", "dt = 0.01", "matlab tf"]);
```

We apply PID control to the above plant system, an electric motor system. We can model the PID control with a discrete equation. The continuous equation of PID control is

$$T(t) = K_P\, e(t) + K_I\, \Sigma\, e(t)\, \Delta t + K_D\, de(t)/dt \quad (\Sigma\text{: integral})$$

where $e(t)$ is the error signal and $T(t)$ is the control signal, that is, the torque signal in the motor system. The corresponding discrete equation is given as follows:

$$T(n) = T(n-1) + K_P\, [e(n)-e(n-1)] + K_I \quad e(n)\, \Delta t + K_D\, [e(n)-2e(n-1)+e(n-2)] / \Delta t$$

The MATLAB program to control the motor speed is given as follows:

```
%
%   PID controller simulation
%
close
dt=0.1; I=1; c=1;  % wheel motor parameters

kp=1; ki=1; kd=0.1;  % PID parameters
```

```
w(1)=0;  w(2)=0;        % initial setting for wheel speed
t(1)=0; t(2)=0;         % initial setting for torque

L=5;             % left motor speed;
r = L*ones(1000,1);     % reference input is step function
r(1:100) = zeros(100,1);        % initial motor speed is zero
e(1)=0; e(2)=0;     %

for n=3:1000,       % we start with n=3 (why?, because of e(n-2))
e(n) = r(n)-w(n-1);
t(n)=t(n-1)+kp*(e(n)-e(n-1))+ki*e(n)*dt+kd*(e(n)-2*e(n-1)+e(n-2))/ dt;
w(n)=w(n-1) + dt/I*(-c*w(n-1) + t(n));
WL=w(n);            % transient angular speed of left motor
end;
plot((1:1000)*dt - dt, r); hold on;
plot((1:1000)*dt - dt, w);
axis([0 100 0 6])
```

The above MATLAB code simulates the transient time response for the left motor speed, when PID control is applied. You need to simulate the right motor speed with separate PID control. Finally, you will obtain the instantaneous motor speed in time course for the left and right motor, respectively. The robot controller will issue the motor commands *(L, R)* of motor speeds to the motor system. The motor commands will vary in time course, depending on sensor information. This mapping from sensor information to motor commands should be coded as the robot control operation (An example of C code is available in the lecture note).

You can set up the initial position and initial head angle on your own choice. The L and R values indicate the reference angular speeds of the motor in the control system. The transient angular speed of left motor is $\boldsymbol{W_L}$ and then the linear velocity of left wheel on a floor should be calculated with $\boldsymbol{V_L} = r\, \boldsymbol{W_L}$, where r is the radius of the robot wheel. The left wheel of the robot moves in a distance of $r\, \boldsymbol{W_L}\, \Delta t$ mm after Δt seconds. We assume the wheel diameter is 5.5 mm and the time step Δt is 0.1 second. Apply the kinematics equation with the transient linear velocities ($\boldsymbol{V_L}$ $\boldsymbol{V_R}$) for left and right wheels to simulate the robotic movement.

Find PID parameters *(Kp, Ki, Kd)* such that the time response (angular speed of motor) has an overdamped response and an underdamped response, respectively. You need to choose appropriate parameters to guarantee the stability. The plant system for angular speed of motors has $I=1$, $c=1$ and so it has a first-order system $1/(s+1)$. Then simulate the robotic movement with a sequence of the motor commands, *(L=0, R=0,), (L=5, R=10), (L=7, R=10), (L=10, R=5), (L=8, R=5)*. Each new command is issued at different time steps – see Table 1. That is, the robot initially stays at the original position and after 10 time steps, the robot starts to move. Then the left & right motors will have transient angular speeds and thus transient linear velocities in the floor space.

Time step	0	10	200	400	600	800
Motor commands	L=0, R=0	L=5, R=10	L=7, R=10	L=10, R=10	L=10, R=5	L=8 R=5

Table 1. A sequence of motor commands (at the given time step, new motor commands start.)

Submissions

All the problems have to use Table 1 which is control commands. Each time step, motor commands are changed as referred Table 1.

(1)Determine PID parameters which shows the lowest steady state error for the wheel speed. Show the angular velocity graph. Compare P control, PI control and PID control. Explain and analyze the result.

(2) Draw a graph of transient angular speeds in time course (from 0 time step to 1000 time steps) for the command set in Table 1, together with the corresponding motor commands in time course, when the PID controller is applied. You need to draw a graph for each case of **overdamped response** and **underdamped response.**

(3) Simulate the robot movement with two wheels. The above experiments test a single wheel motor. Here, we test two wheel speeds of a robot. Draw the trajectories of robotic movements in the floor space (without any arena boundary), following the above kinematic simulation. Discuss what kind of trajectories can be observed for **the overdamped** response and the **underdamped response** with the PID control and which one is more desirable.

You need to write a report for the above results and your program code (11pt font is recommended, and please follow the instructions given in the course syllabus) and add the source codes you have modified and the result at the end. You should submit **ZIP file which composed with MATLAB codes and report** in the YSCEC website.

(Appendix) basic robot simulation code

```
close all;
head =90*pi/180 ; x=0; y=0;  % starting position and heading angle

diameter = 55;   radius = diameter /2;  % robot diameter
wdiameter=5.5;   wradius = wdiameter /2;%wheel diameter
B = 50;  % distance between two wheels

t = 0:0.1:2*pi+0.2; % to draw robot body
dt=0.1;
I=1; c=1;  % wheel motor parameters
kp=1; ki=1; kd=0.1;  % PID parameters

scrsz = get(0,'ScreenSize');
figure('Position',[100 100 scrsz(3)*0.8 scrsz(4)*0.8])

time=0;
Numberofloop=1000;
wL=0;%  anguler velocity of left wheel
wR=0;%  anguler velocity of right wheel
tL=0;
tR=0;
w = 0;
savex=zeros(1,Numberofloop);
savey=zeros(1,Numberofloop);
saveR=zeros(1,Numberofloop);
savehead = zeros(1,Numberofloop);
savewL=zeros(1,Numberofloop);
savewR=zeros(1,Numberofloop);
for N=1:1:Numberofloop

    rx = x + radius * cos(t); ry = y + radius * sin(t); %for robot
drawing
    subplot(2,2,1),plot(rx, ry)% draw robot body
    axis([-600 600 -600 600])
    daspect([1 1 1])
    line([x x+radius*cos(head)], [y y+radius*sin(head)])
    subplot(2,2,2),plot(x, y,'.');% draw robot trajectory
    axis([-600 600 -600 600]),hold on;
    daspect([1 1 1])
    subplot(2,2,3),plot(time,w ,'.');% draw distance between robot and
center
    axis([0 Numberofloop*dt -pi/2 pi/2]),hold on;
    subplot(2,2,4),plot(time, wL,'x'),hold on % draw angular velocity
    plot(time, wR,'o'),
    axis([0 Numberofloop*dt 0 15]),hold on;

    %%%%
    %you may determine desired angular velocity here.
    dwL = 5;
    dwR = 10;
    %%%%
```

```
    %%%%
    %you may determine torque here.
    %%%%

    %%%%
    %you may apply torque to robot system here
    wL = dwL;
    wR = dwR;
    %%%%
      VL=wL*wradius;%wheel speed
      VR=wR*wradius;%wheel speed

    if(VL==VR)
        x=x+VL*dt*cos(head);
        y=y+VL*dt*sin(head);
    else
        w=(VR-VL)/B;
    R=(B*(VR+VL))/((VR-VL)*2);
    x=R*sin(w*dt+head)+x-R*sin(head);
    y=-R*cos(w*dt+head)+y+R*cos(head);
    head=mod(head+w*dt,2*pi);
    end

    pause(0.1);   %
    savex(1,N)=x;
    savey(1,N)=y;
    savehead(1,N)=head;
    savewL(1,N)=wL;
    savewR(1,N)=wR;
    time=time+dt;
end
```