
Table of Contents

Homework 2	1
2a	1
2b	2
3	4

Homework 2

Bailey Waterman, Keshuai Xu

```
pu = get(gcf, 'PaperUnits');  
pp = get(gcf, 'PaperPosition');  
set(gcf, 'Units', pu, 'Position', pp)
```

2a

```
clear variables; close all; clc;  
  
v_e = [10 0 0]'; % units  
R_i_e = rotx(deg2rad(13))*roty(deg2rad(15))*rotz(deg2rad(10));  
  
v_i = R_i_e \ v_e % units
```

```
v_i =  
  
    9.9998  
   -0.0305  
    0.0457
```

2b

```
function euler321_angles_dot = rotational_kinematics(t,  
    euler321_angles, omega_iee)  
  
psi_yaw = euler321_angles(1);  
theta_pitch = euler321_angles(2);  
phi_roll = euler321_angles(3);  
  
H_321 = [-sin(theta_pitch) 0 1;  
    sin(phi_roll)*cos(theta_pitch) cos(phi_roll) 0;  
    cos(phi_roll)*cos(theta_pitch) -sin(phi_roll) 0];  
  
euler321_angles_dot = H_321 \ omega_iee;  
end  
  
clear variables; close all; clc;  
  
%initial conditions  
euler321_angles_initial = [10; 15; 13] * pi/180; %radians  
time_span = [0 2]; %seconds  
omega_iee = [2; 0; 0] * pi/180; %radians/second  
  
%solve ODE  
[t_sim, euler321_angles_sim] = ...  
ode45(@(t, y)rotational_kinematics(t, y, omega_iee), time_span,  
    euler321_angles_initial);  
  
%initial conditions same as last ones  
euler321_angles_initial = euler321_angles_sim(end, :); %rad  
time_span = [2 5]; %s  
omega_iee = [1; 3; 0] * pi/180; %rad/s  
  
%solve ODE  
[t_sim_2, euler321_angles_sim_2] = ...  
ode45(@(t, y)rotational_kinematics(t, y, omega_iee), time_span,  
    euler321_angles_initial);  
  
%initial conditions same as last ones  
euler321_angles_initial = euler321_angles_sim_2(end, :); %rad
```

```
time_span = [5 8]; %s
omega_iee = [0; 0; 1;] * pi/180; %rad/s

%solve ODE
[t_sim_3, euler321_angles_sim_3] = ...
ode45(@(t, y)rotational_kinematics(t, y, omega_iee), time_span,
    euler321_angles_initial);

%initial conditions same as last ones
euler321_angles_initial = euler321_angles_sim_3(end, :); %rad
time_span = [8 10]; %s
omega_iee = [1; 4; 3;] * pi/180; %rad/s

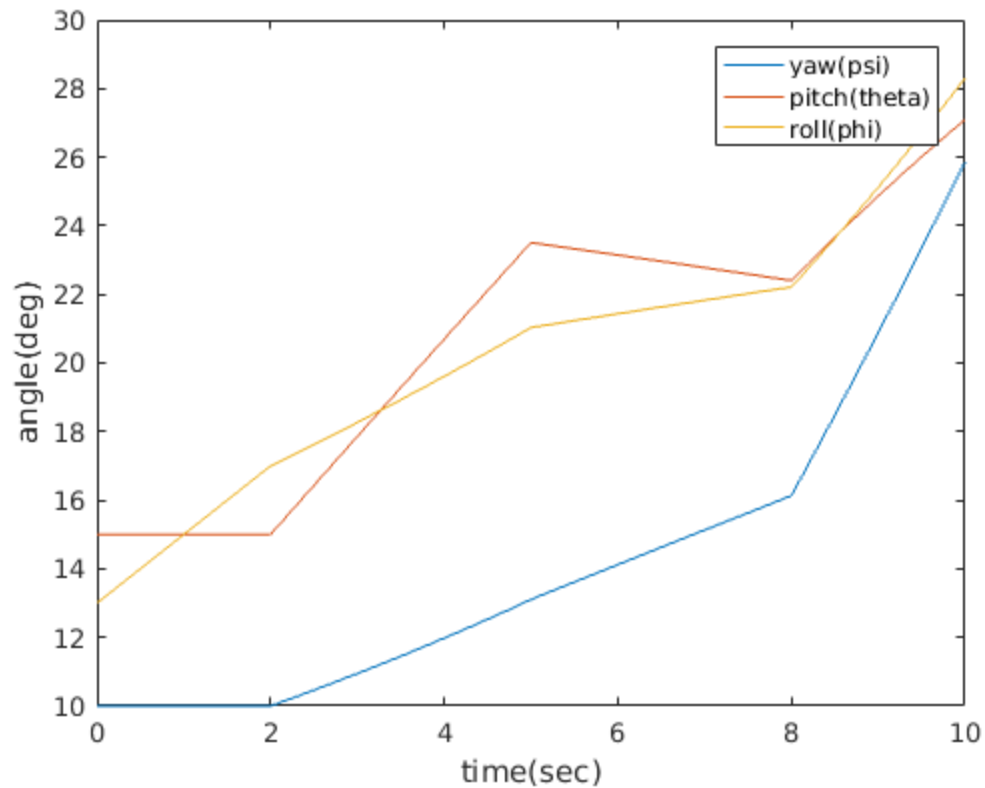
%solve ODE
[t_sim_4, euler321_angles_sim_4] = ...
ode45(@(t, y)rotational_kinematics(t, y, omega_iee), time_span,
    euler321_angles_initial);

euler321_angles_degrees_1 = euler321_angles_sim * 180/pi;
euler321_angles_degrees_2 = euler321_angles_sim_2 * 180/pi;
euler321_angles_degrees_3 = euler321_angles_sim_3 * 180/pi;
euler321_angles_degrees_4 = euler321_angles_sim_4 * 180/pi;

time = [t_sim; t_sim_2; t_sim_3; t_sim_4;];
euler321_angles_degrees = [euler321_angles_degrees_1;
    euler321_angles_degrees_2; euler321_angles_degrees_3;
    euler321_angles_degrees_4;];

plot(time, euler321_angles_degrees)

xlabel('time(sec)')
ylabel('angle(deg)')
legend('yaw(psi)', 'pitch(theta)', 'roll(phi)')
```



3

```
clear variables; close all; clc;
% from the derivation on the paper, v_dot_i is a piecewise function of
t.
v_dot_i = @(t) [hw2_a_t(t)*cos(hw2_psi(t)) -
    hw2_psi_dot(t)*hw2_v(t)*sin(hw2_psi(t));
    hw2_a_t(t)*sin(hw2_psi(t)) +
    hw2_psi_dot(t)*hw2_v(t)*cos(hw2_psi(t));
    0];
% in odefun and initial conditon y0,
% y(1:3, :) = p_i (position in m)
% y(4:6, :) = v_i (velocity in m/s)
odefun = @(t,y) [y(4:6, :); v_dot_i(t)];
y0 = [10 0 0 5 0 0]';
warning('off','all')
[t_sim, y_sim] = ode45(odefun, [0 40], y0);
warning('on','all')
figure();
plot(t_sim, y_sim(:, 1:3));
xlabel('time(s)')
ylabel('position(m)')
legend({'$\hat{i}$', '$\hat{j}$', '$\hat{k}$'}, 'Interpreter', 'latex')
figure();
plot(t_sim, y_sim(:, 4:6));
```

```

xlabel('time(s)')
ylabel('velocity(m/s)')
legend({'$\hat{i}$', '$\hat{j}$', '$\hat{k}$'}, 'Interpreter', 'latex')
% at t=40 in I frame,
position = y_sim(end, 1:3)' % m
velocity = y_sim(end, 4:6)' % m/s

```

```
position =
```

```

    977.1305
   -462.6104
           0

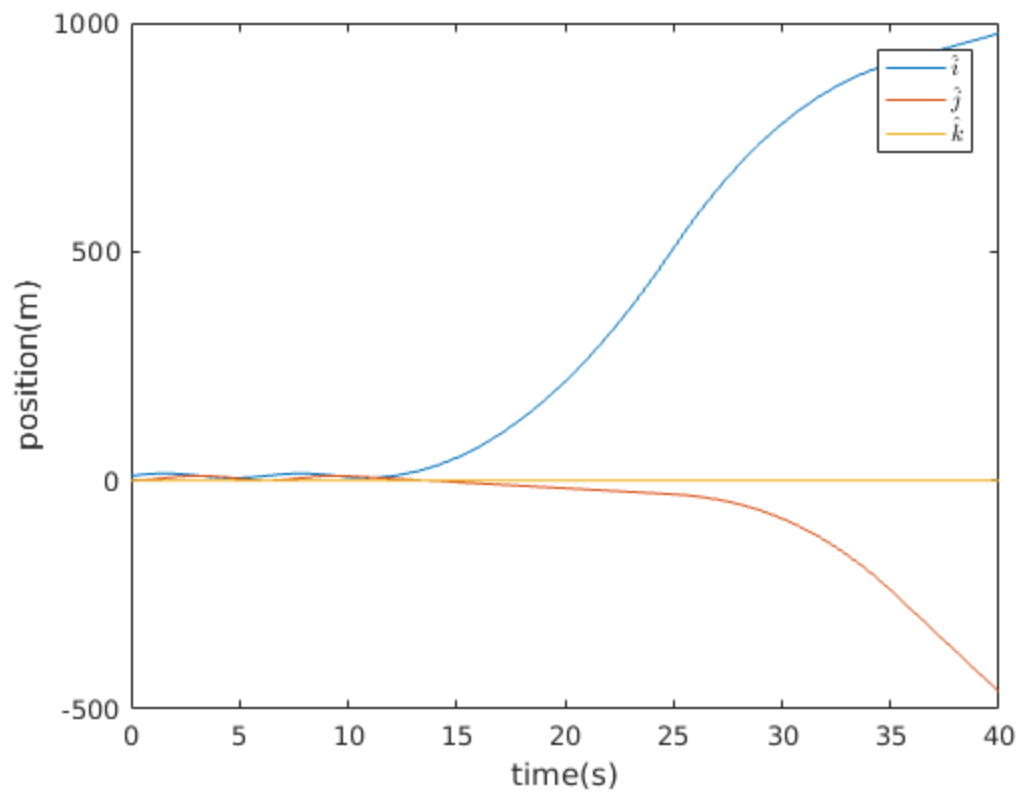
```

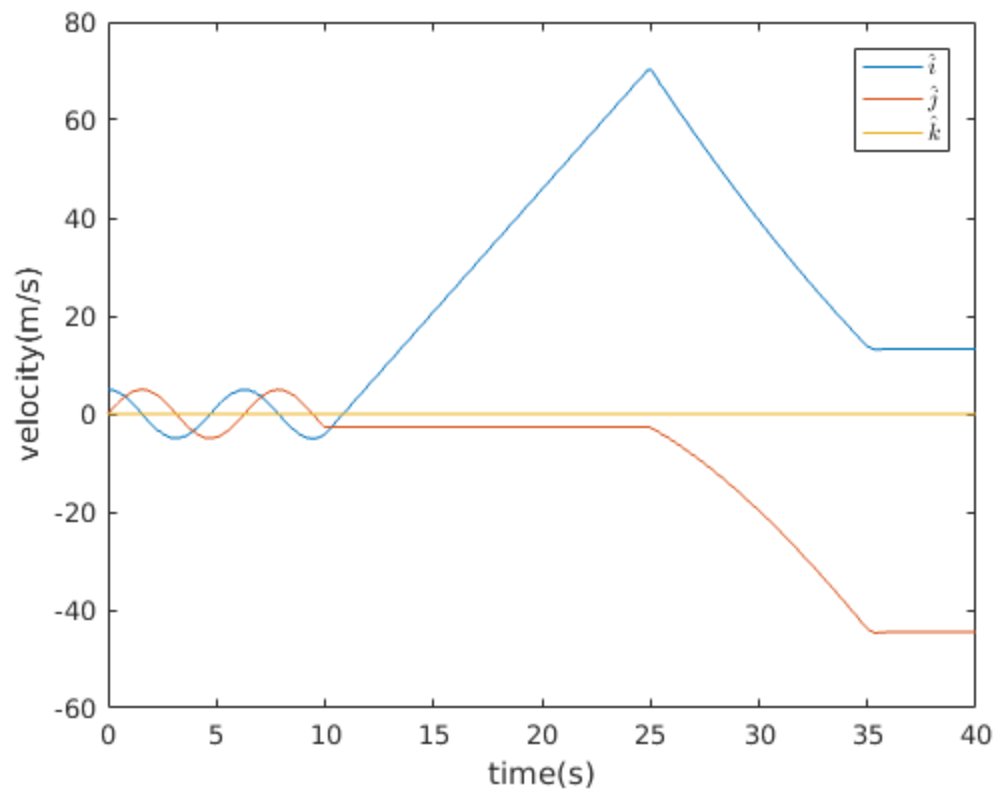
```
velocity =
```

```

    13.2583
   -44.5351
           0

```





Published with MATLAB® R2016a