GNC midterm

Table of Contents

Keshuai Xu

Problem 1

1. 3.1623e-14 W*m^(-2)

2. True

3. True

4. False

5. False

6. [0 0 0]'

7. approx [0 0 2*pi/86400]

8. 6356.752314245 km

9. Proper acceleration in an inertial coordinate system

10. [0 0 0]'

Problem 2

Variable naming conventions: R_subscript_superscript x_d is dx/dt. x_dd is d(x_d)/dt.

```
clear variables; close all;
syms psi theta phi psi_d theta_d phi_d real
```

2.1

```
% x rotation psi
```

```
R_i_a = [1 0 0;
         0 cos(psi) sin(psi);
         0 -sin(psi) cos(psi)]

% y rotation theta
R_a_b = [cos(theta) 0 -sin(theta);
         0 1 0;
         sin(theta) 0 cos(theta)]

% z rotation phi
R_b_e = [cos(phi) sin(phi) 0;
         -sin(phi) cos(phi) 0;
         0 0 1]
```

```
R_i_a =

[ 1,         0,        0]
[ 0,  cos(psi), sin(psi)]
[ 0, -sin(psi), cos(psi)]


R_a_b =

[ cos(theta), 0, -sin(theta)]
[          0, 1,           0]
[ sin(theta), 0,  cos(theta)]


R_b_e =

[  cos(phi), sin(phi), 0]
[ -sin(phi), cos(phi), 0]
[         0,        0, 1]
```

2.2

```
R_i_e = R_b_e * R_a_b * R_i_a;
```

2.3

$$\vec{\omega}_{ie}^{e} = R_b^e R_a^b \vec{\omega}_{ia}^{a} + R_b^e \vec{\omega}_{ab}^{b} + I \vec{\omega}_{be}^{e}$$

2.4

```
omega_ia_a = [psi_d 0 0]'
omega_ab_b = [0 theta_d 0]'
omega_be_e = [0 0 phi_d]'
```

```
omega_ia_a =
```

```
 psi_d
     0
     0


omega_ab_b =

       0
 theta_d
       0


omega_be_e =

     0
     0
 phi_d
```

2.5

```
omega_ie_e = R_b_e * R_a_b * omega_ia_a + R_b_e * omega_ab_b +
 omega_be_e

% ANSWER - the part to fill in the blank
H_123_e = [cos(phi)*cos(theta) sin(phi) 0;
           -cos(theta)*sin(phi) cos(phi) 0;
           sin(theta) 0 1]
```

```
omega_ie_e =

 theta_d*sin(phi) + psi_d*cos(phi)*cos(theta)
 theta_d*cos(phi) - psi_d*cos(theta)*sin(phi)
                     phi_d + psi_d*sin(theta)


H_123_e =

[  cos(phi)*cos(theta), sin(phi), 0]
[ -cos(theta)*sin(phi), cos(phi), 0]
[           sin(theta),        0, 1]
```

2.6

ANSWER - the part to fill in the blank

$$\vec{\omega}_{ie}^{i} = \underline{(\cos\theta)^{-1}} B \vec{\omega}_{ie}^{e}$$

```
% intermediate step and verify 2.5
```

```
omega_ie_i = H_123_e \ omega_ie_e

% the whole thing
inv_H123_e = simplify(inv(H_123_e))

% ANSWER - B part of the answer
B = simplify(inv_H123_e * cos(theta))
```

```
omega_ie_i =

   psi_d
 theta_d
   phi_d


inv_H123_e =

[               cos(phi)/cos(theta),             -sin(phi)/cos(theta),
 0]
[                          sin(phi),                         cos(phi),
 0]
[ -(cos(phi)*sin(theta))/cos(theta), (sin(phi)*sin(theta))/cos(theta),
 1]


B =

[              cos(phi),           -sin(phi),          0]
[  cos(theta)*sin(phi), cos(phi)*cos(theta),          0]
[ -cos(phi)*sin(theta), sin(phi)*sin(theta), cos(theta)]
```

2.7

quantity measured by gyro: $\vec{\omega}_{ie}^{e}$

```
% convert from symbolic expression to function
inv_H123_e_fun = matlabFunction(inv_H123_e, 'Vars', [psi theta phi]');

% euler123_i = [psi theta phi]'
omega_ie_i_fun = @(euler123_i, omega_ie_e)
 inv_H123_e_fun(euler123_i(1), euler123_i(2), euler123_i(3)) *
 omega_ie_e;

[t_sim_27, y_sim_27] = ode45(@(t, y) omega_ie_i_fun(y, deg2rad([0 0.25
 0.75]')), [0 20], deg2rad([5 10 15]'));

% ANSWER [psi theta phi]' deg
rad2deg(y_sim_27(end, :)')
```

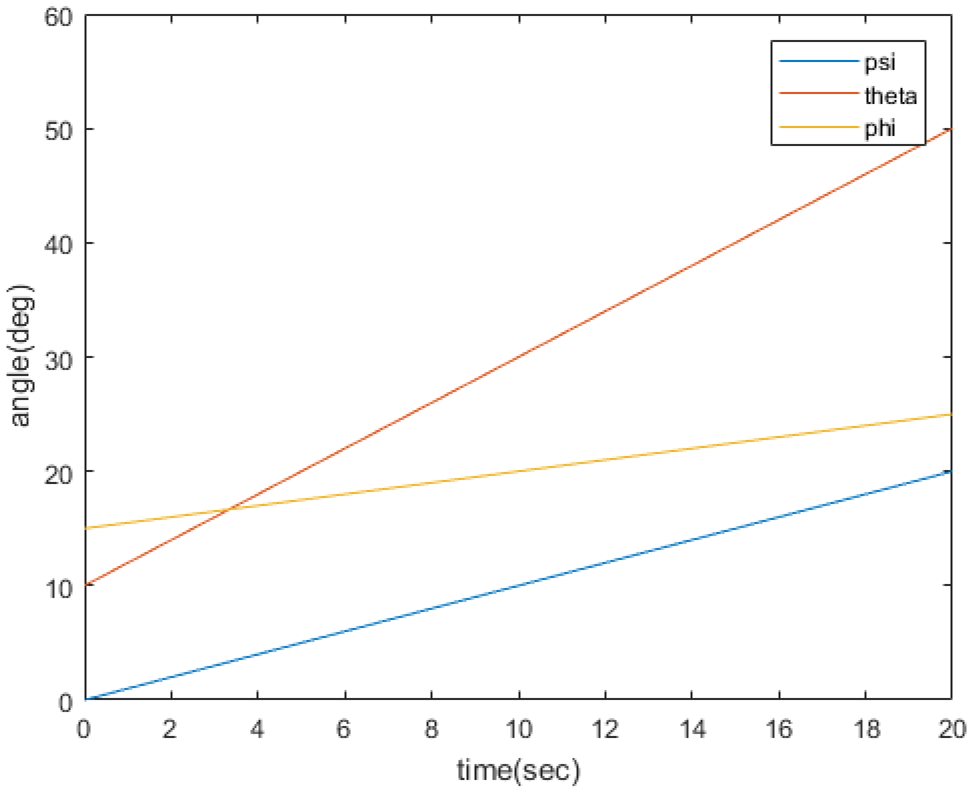
```
ans =
```

```
    3.0301
   14.5995
   30.4295
```

2.8

```
DATA28 = load('gyrodata_p2_8_2017.mat');

% linear interpolation of the gyro data
omega_ie_e_fun = @(t) interp1(DATA28.time_pts', DATA28.omega_iee',
 t)';

[t_sim28, y_sim28] = ode45(@(t, y) omega_ie_i_fun(y,
 omega_ie_e_fun(t)), DATA28.time_pts, [DATA28.psi_0 DATA28.theta_0
 DATA28.phi_0]');

figure();
plot(t_sim28, rad2deg(y_sim28));
xlabel('time(sec)');
ylabel ('angle(deg)');
legend ('psi','theta','phi');

% ANSWER [psi theta phi]' deg
rad2deg(y_sim28(end, :)')
```

```
ans =

   20.0000
   50.0000
   25.0000
```



3

```
clear variables
syms psi gamma psi_d gamma_d airspeed airspeed_dot real

R_i_a = [cos(psi) sin(psi) 0;
         -sin(psi) cos(psi) 0;
         0 0 1];
R_a_e = [cos(gamma) 0 -sin(gamma);
         0 1 0;
         sin(gamma) 0 cos(gamma)];

omega_ia_a = [0 0 psi_d]';
omega_ae_e = [0 gamma_d 0]';
omega_ie_e = R_a_e * omega_ia_a + omega_ae_e;

% @(gamma_d,gamma,psi_d)[-psi_d.*sin(gamma);gamma_d;psi_d.*cos(gamma)]
omega_ie_e_fun = matlabFunction(omega_ie_e);

R_i_e = R_a_e * R_i_a;

% @(gamma,psi)
R_i_e_fun = matlabFunction(R_i_e);
```

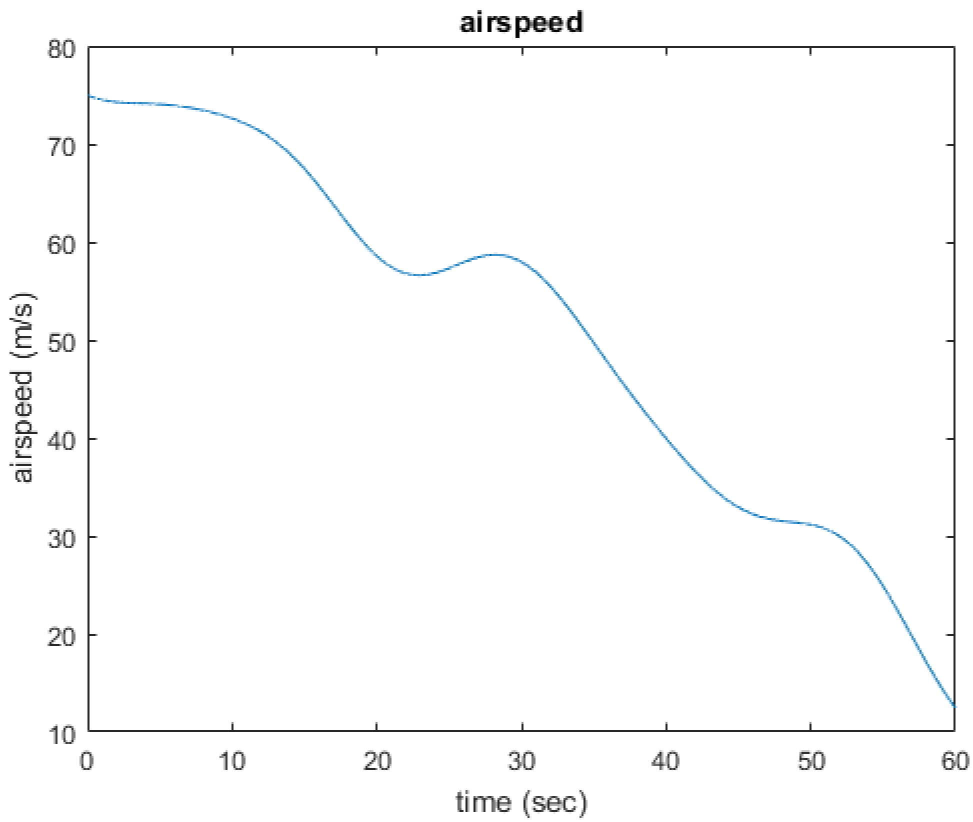
```
v_e = [airspeed 0 0]';
v_dot_e = [airspeed_dot 0 0]';
v_i = R_i_e' * v_e;

accel_reading_inertial_sym = v_dot_e + cross(omega_ie_e, v_e)
% @(airspeed,airspeed_dot,gamma_d,gamma,psi_d)
accel_reading_inertial_sym_fun =
 matlabFunction(accel_reading_inertial_sym);


psi_t0 = deg2rad(45);
gamma_t0 = deg2rad(2);
psi_dot_t0 = deg2rad(0);
gamma_dot_t0 = deg2rad(1.5);
airspeed_t0 = 75; % m/s
% assume there's no tangential acceleration at this moment, because
 it's not given in the problem
airspeed_dot_t0 = 0;

% ANSWER inertial acceleration (no gravity) from accelerometer
accel_reading_inertial_num =
 accel_reading_inertial_sym_fun(airspeed_t0,airspeed_dot_t0,gamma_dot_t0,gamma_t0,

g_i = [0 0 9.81]';
g_e = R_i_e_fun(gamma_t0, psi_t0) * g_i;

% ANSWER accelerometer reading with gravity
accel_reading_num = accel_reading_inertial_num - g_e


accel_reading_inertial_sym =

              airspeed_dot
 airspeed*psi_d*cos(gamma)
         -airspeed*gamma_d


accel_reading_inertial_num =

         0
         0
   -1.9635


accel_reading_num =

    0.3424
         0
  -11.7675
```

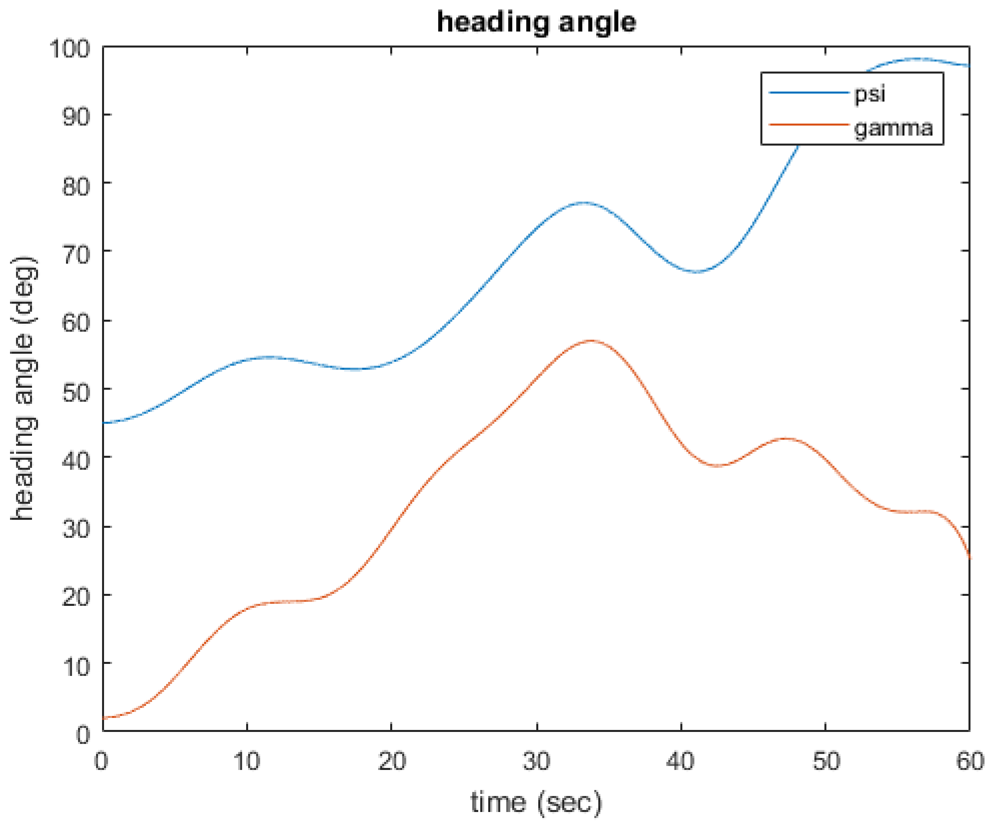
4

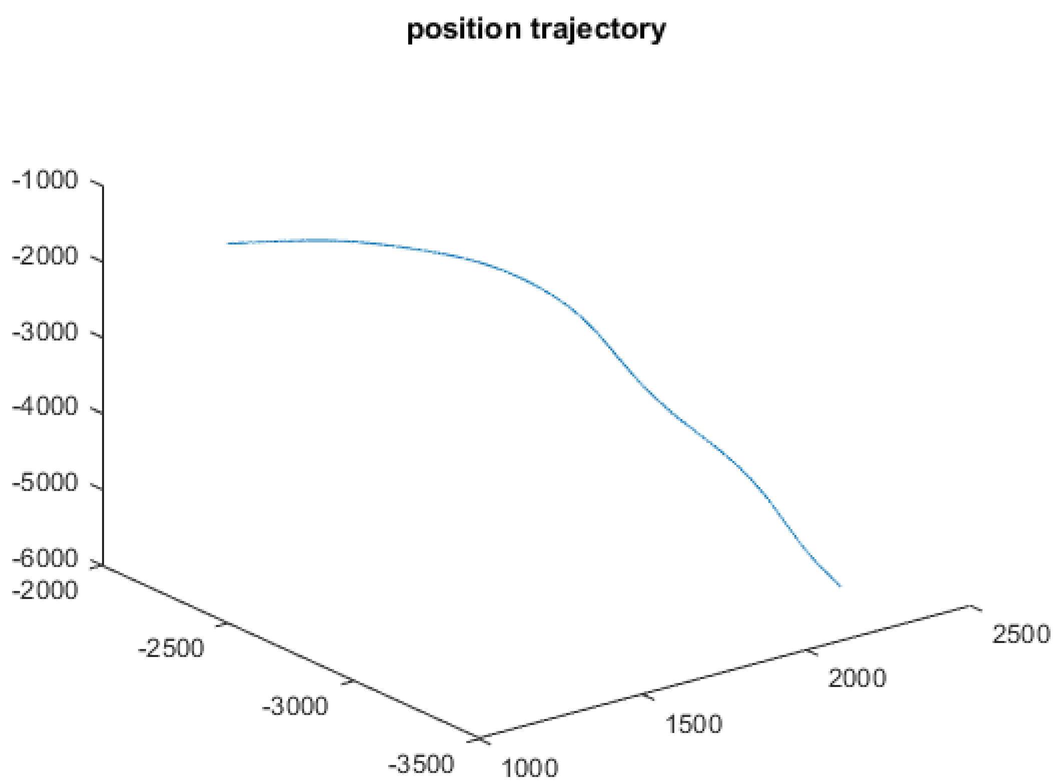
```
function y_dot = odefun_4 (t, y, R_i_e_fun, DATA4)
    airspeed = y(1,:);
    psi = y(2,:);
    gamma = y(3,:);
    accel_reading = interp1(DATA4.time_pts', DATA4.accel_readings',
 t)';
    p_ddot_t = R_i_e_fun(gamma, psi)' * accel_reading;

    y_dot = [accel_reading(1,:); % tangential accel
        accel_reading(2,:)/(airspeed * cos(gamma)); % psi_dot
        accel_reading(3,:)/(-airspeed); % gamma_dot
        y(7:9,:); % p_dot_t
        p_ddot_t]; % p_dot_dot_t
end
```

```
DATA4 = load('acceldata_p4_2017.mat');

gamma_psi_0 = deg2rad([45; 2]);
p_dot_e_0 = [75 0 0]';


% y(9x1) = [airspeed psi gamma p_t(3x1) p_dot_t(3x1)]'
y0 = [p_dot_e_0(1); gamma_psi_0; DATA4.posn_0;
 R_i_e_fun(gamma_psi_0(1), gamma_psi_0(2))' * p_dot_e_0];

[t_4, y_4] = ode45(@(t,y)
 odefun_4(t,y,R_i_e_fun,DATA4),DATA4.time_pts,y0);

figure();
plot(t_4, y_4(:,1));
xlabel('time (sec)');
ylabel ('airspeed (m/s)');
title('airspeed');

figure();
plot(t_4, rad2deg(y_4(:,2:3)));
xlabel('time (sec)');
ylabel ('heading angle (deg)');
legend ('psi','gamma');
title('heading angle');

figure();
plot3(y_4(:,4),y_4(:,5),y_4(:,6));
title('position trajectory');

% ANSWER final position in m
p_t60 = y_4(end,4:6)'
```

```
p_t60 =

   1.0e+03 *

    2.1716
   -3.4122
   -5.4986
```




heading angle
heading angle (deg)
time (sec)
psi
gamma
100
90
80
70
60
50
40
30
20
10
0
0
10
20
30
40
50
60


position trajectory
-1000
-2000
-3000
-4000
-5000
-6000
-2000
-2500
-3000
-3500
1000
1500
2000
2500

5

```
clear variables
syms psi theta alpha beta real

% x rotation psi
R_i_a = [1 0 0;
         0 cos(psi) sin(psi);
         0 -sin(psi) cos(psi)];

% y rotation theta
R_a_b = [cos(theta) 0 -sin(theta);
         0 1 0;
         sin(theta) 0 cos(theta)];

% z rotation alpha
R_i_c = [cos(alpha) sin(alpha) 0;
         -sin(alpha) cos(alpha) 0;
         0 0 1];

% x rotation beta
R_c_d = [1 0 0;
         0 cos(beta) sin(beta);
         0 -sin(beta) cos(beta)];

k_i = [0 0 1]';

k_b = R_a_b * R_i_a * k_i;
k_d = R_c_d * R_i_c * k_i;

y = k_b - k_d;

% answer is to find [psi theta alpha beta] that makes norm(y) = 0
% ANSWER: y2 == 0
y2 = simplify(y(1)^2 + y(2)^2 + y(3)^2)
```

```
y2 =

(cos(beta) - cos(psi)*cos(theta))^2 + (sin(beta) - sin(psi))^2 +
 cos(psi)^2*sin(theta)^2
```

Published with MATLAB® R2017a