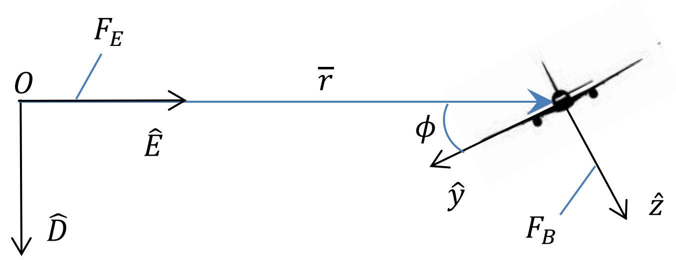
ASEN 3128 Assignment 1--Solutions

Due: Monday, January 29 at Noon, on the Dropbox on D2L.

An aircraft is flying in a level right turn, with a turn radius of 1km about a point *O*, and a speed relative to frame F_E of 100m/s. At a particular instant of time, the situation is as shown in the figure below.



Parts 1-8 below refer to this situation at the instant of time shown. In each case, provide a quantitative description as well as descriptive terminology for that entity. Wherever possible, also provide vector components using the textbook's notation.

1) $\bar{r}$, $\bar{r}_E$, $\bar{r}_B$

$\bar{r}$ is a position vector from the origin of the of the Earth-Fixed Frame to the center of gravity of the aircraft.

$$\bar{r} = 1000\ \hat{E}\ [m]; \qquad |\bar{r}| = 1000\ [m]$$

$\bar{r}_E$ is the inertial coordinate representation of the position vector $\bar{r}$. i.e. represented in the frame $F_E = (O, \hat{N}, \hat{E}, \hat{D})$

$$\bar{r}_E \triangleq \begin{bmatrix} x_E \\ y_E \\ z_E \end{bmatrix} = \begin{bmatrix} 0 \\ 1000 \\ 0 \end{bmatrix}\ [m]$$

$\bar{r}_B$ is the body-fixed coordinate representation of the position vector $\bar{r}$. i.e. represented in the frame $F_B = (O, \hat{x}, \hat{y}, \hat{z})$. This is easily found by expressing $\hat{E}$ in body coordinates:

$$\bar{r}_B \triangleq \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0 \\ -1000\cos(\phi) \\ 1000\sin(\phi) \end{bmatrix}\ [m]$$

2) $\overline{V}^E = \frac{d}{dt}^E \; \overline{r}$, $\overline{V}_E^E$, $\overline{V}_B^E$

$\bar{V}^E$ is the inertial velocity vector of the aircraft, tangent to the trajectory of motion as seen by the inertial frame:

$$\bar{V}^E \triangleq \frac{d}{dt}^E \; \bar{r} = -100 \, \widehat{N} \; [m/s]$$

$\bar{V}_E^E$ is the inertial velocity vector represented in inertial coordinates. i.e. represented in the frame F_E

$$\bar{V}_E^E \triangleq \begin{bmatrix} \dot{x}_E \\ \dot{y}_E \\ \dot{z}_E \end{bmatrix} = \begin{bmatrix} -100 \\ 0 \\ 0 \end{bmatrix} \; [m/s]$$

$\bar{V}_B^E$ is the inertial velocity vector represented in body coordinates. i.e. represented in the frame F_B

$$\bar{V}_B^E \triangleq \begin{bmatrix} u^E \\ v^E \\ w^E \end{bmatrix} = \begin{bmatrix} 100 \\ 0 \\ 0 \end{bmatrix} \; [m/s]$$

3) $\overline{\omega}^{EB}$, $\overline{\omega}_E^{EB}$, $\overline{\omega}_B^{EB}$

$\bar{\omega}^{EB}$ is the angular velocity vector of the aircraft (body frame relative to the inertial frame). The aircraft is rotating about the $\widehat{D}$ direction as it orbits on the circle, with magnitude $\frac{|\bar{V}^E|}{|\bar{r}|}$

$$\bar{\omega}^{EB} = \frac{100 \, \frac{m}{s}}{1000 \, m} \widehat{D} = 0.1 \, \widehat{D} \; [rad/s]$$

$\bar{\omega}_E^{EB}$ is the angular velocity vector represented in inertial coordinates. i.e. represented in the frame F_E

$$\bar{\omega}_E^{EB} = \begin{bmatrix} 0 \\ 0 \\ 0.1 \end{bmatrix} \; [rad/s]$$

$\bar{\omega}_B^{EB}$ is the angular velocity vector represented in body coordinates. i.e. represented in the frame F_B

$$\bar{\omega}_B^{EB} \triangleq \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 0 \\ 0.1 \sin(\phi) \\ 0.1 \cos(\phi) \end{bmatrix} \; [rad/s]$$

4) $\frac{d}{dt}^B \ \bar{r}$, $\left(\frac{d}{dt}^B \ \bar{r}\right)_E$, $\left(\frac{d}{dt}^B \ \bar{r}\right)_B$

$\frac{d}{dt}^B \ \bar{r}$ is the body frame derivative of the position vector $\bar{r}$. Geometrically, as the plane translates around the circular flight path, the position vector $\bar{r}$ always looks the same in the body-fixed frame (coming in over the right wing at angle ϕ). Hence

$$\frac{d}{dt}^B \ \bar{r} = 0 \ \hat{N} \ [m/s]$$

$\left(\frac{d}{dt}^B \ \bar{r}\right)_E$ is the body frame derivative of the position vector $\bar{r}$ represented in inertial coordinates, i.e. represented in the frame F_E

$$\left(\frac{d}{dt}^B \ \bar{r}\right)_E = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \ [m/s]$$

$\left(\frac{d}{dt}^B \ \bar{r}\right)_B$ is the body frame derivative of the position vector $\bar{r}$ represented in body coordinates, i.e. represented in the frame F_B

$$\left(\frac{d}{dt}^B \ \bar{r}\right)_B = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \ [m/s]$$

5) $\left(\frac{d}{dt}^E \ \overline{V}^E\right)_E$, $\left(\frac{d}{dt}^B \ \overline{V}^E\right)_B$

$\left(\frac{d}{dt}^E \ \overline{V}^E\right)_E$ is the inertial frame derivative of the inertial velocity, expressed in inertial coordinates. Also called the inertial acceleration, represented in inertial coordinates. For circular motion, this acceleration is centripetal (directed along $-\hat{E}$) with magnitude $\frac{|\bar{V}^E|^2}{|\bar{r}|}$:

$$\left(\frac{d}{dt}^E \ \overline{V}^E\right)_E = \begin{bmatrix} 0 \\ -10 \\ 0 \end{bmatrix} \left[\frac{m}{s^2}\right]$$

$\left(\frac{d}{dt}^B \ \overline{V}^E\right)_B$ is the body frame derivative of the inertial velocity expressed in body coordinates. This is **not** the inertial acceleration, but a mixed inertial/body second derivative. Since the velocity vector $\overline{V}^E$ is always directed along $\hat{x}$ for circular motion, the inertial velocity is not changing with respect to the body frame, hence

$$\left(\frac{d}{dt}^B \overline{V}^E\right)_B = \frac{d}{dt}\overline{V}_B^E = \begin{bmatrix}\dot{u}^E\\ \dot{v}^E\\ \dot{w}^E\end{bmatrix} = \begin{bmatrix}0\\0\\0\end{bmatrix}\left[\frac{m}{s^2}\right]$$

6) Calculate $\frac{d}{dt}^E \overline{V}^E$ using the velocity rule

$$\frac{d}{dt}^E \overline{V}^E = \frac{d}{dt}^B \overline{V}^E + \overline{\omega}^{EB} \times \overline{V}^E$$

$$\frac{d}{dt}^E \overline{V}^E = \overline{0} + 0.1\,\widehat{D} \times (-100)\widehat{N} = -10\,\widehat{E}\,\left[\frac{m}{s^2}\right]$$

7) $\overline{f}$, $\overline{f}_E$, $\overline{f}_B$

$\bar{f}$ is the net force vector acting on the aircraft. Using Newton's law with aircraft mass m

$$\bar{f} = m\,\frac{d}{dt}^E \overline{V}^E$$

$$\bar{f} = -10\,m\,\hat{E}\,[N]$$

$\bar{f}_E$ is the net force vector represented in inertial coordinates:

$$\bar{f}_E = \begin{bmatrix}0\\-10m\\0\end{bmatrix}[N]$$

$\bar{f}_B$ is the net force vector represented in body coordinates

$$\bar{f}_B = \begin{bmatrix}0\\10\,m\,cos\phi\\-10\,m\,sin\phi\end{bmatrix}[N]$$

8) In the above situation, suppose there is a prevailing wind with a N-component of 10 m/s, an E-component of 20 m/s, and a D-component of -5 m/s. What is the relative wind vector $\overline{V}$? What is $\overline{V}_B = \begin{bmatrix}u\\v\\w\end{bmatrix}$?

$$\overline{V}_E^E = \overline{W}_E + \overline{V}_E$$

$$\overline{V}_E^E = \begin{bmatrix}-100\\0\\0\end{bmatrix}[m/s] \qquad \overline{W}_E = \begin{bmatrix}10\\20\\-5\end{bmatrix}[m/s]$$

$$\bar{V}_E = \bar{V}_E^E - \bar{W}_E = \begin{bmatrix} -100 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 10 \\ 20 \\ -5 \end{bmatrix}$$

$$\bar{V}_E = \begin{bmatrix} -110 \\ -20 \\ 5 \end{bmatrix} \; [m/s]$$

$$\bar{V} = -110\,\hat{N} - 20\,\hat{E} + 5\,\hat{D}$$

To get $\bar{V}_B$ we need to represent $\bar{V}$ in body frame coords. This is easily done by noting

$$\hat{N} = -\hat{x}$$

$$\hat{E} = -\cos\phi\,\hat{y} + \sin\phi\,\hat{z}$$

$$\hat{D} = \sin\phi\,\hat{y} + \cos\phi\,\hat{z}$$

So

$$\bar{V} = 110\,\hat{x} - 20(-\cos\phi\;\hat{y} + \sin\phi\;\hat{z}) + 5(\sin\phi\,\hat{y} + \cos\phi\,\hat{z})$$

and

$$\bar{V}_B = \begin{bmatrix} 110 \\ 20\cos\phi + 5\sin\phi \\ -20\sin\phi + 5\cos\phi \end{bmatrix} \; [m/s]$$

Another approach is to use the DCM:

$$\bar{V}_B = L_{BE}\bar{V}_E$$

but it is often more tedious to calculate all the terms in L_{BE}.

9) Construct a simulation of the translational dynamics of an aircraft, where the forces on the body are not a function of the body attitude, but include aerodynamic drag and gravity. Use this to model a golf ball, with mass of 50 g, diameter 3.0 cm, and coefficient of drag of 0.5. Begin the simulation with an initial position at the origin of the inertial frame, with an initial velocity of 20 m/s upward and 20 m/s East.

```
% ASEN 3128 - Homework 1, Problem 9
%
% Translational dynamics simulation for an aircraft
% Attitude-independent drag and gravity forces
clc; clear; close all;

% Parameters
```

```
g = 9.81; % acceleration of gravity [m/s^2]
rho = 1.0; % air density [kg/m^3]

% Ode45 Set Up
options = odeset('Events',@ground_event);

% Golf ball parameters
m = 0.05; % mass [kg]
rho_lift = 0; % density of lifting gas [kg/m^3]
vol = 0.0; % volume displaced by lifting gas [m^3]
cd = 0.5; % coefficient of drag
d = 0.03; % golf ball diameter [m]
S = (d/2)^2*pi; % Planform area [m^2]
IC = [0; 0; 0; 0; 20; -20]; % Initial conditions
tspan = 0:0.01:5;
```

Simulation

```
% Wind (direction air is moving relative to the inertial frame)
W = [20, 0, 0]; % wind velocity [North, East, Down] [m/s]

% State: y =
%   [x_E,       % North component of inertial position [m]
%    y_E,       % East component of inertial position [m]
%    z_E,       % Down component of inertial position [m]
%    xdot_E,    % North component of inertial velocity [m/s]
%    ydot_E,    % East component of inertial velocity [m/s]
%    zdot_E]    % Down component of inertial velocity [m/s]
p = [m, g, rho, vol, cd, S, W];

% Simulate with ode45
[t,y] = ode45(@(t,y) translation_model(t, y, p), tspan, IC, options);

figure()
plot(t, y)
xlabel('Time, [s]')
ylabel('State Components')
legend({'$x_E$', '$y_E$', '$z_E$', '$\dot{x_E}$', '$\dot{y_E}$', '$\dot{z_E}$'},...
    'Interpreter', 'latex', 'Location', 'NorthWest')

figure()
plot3(y(:,2), y(:,1), -y(:,3))
xlabel('E position, [m]')
ylabel('N position, [m]')
zlabel('Altitude, [m]')
axis equal
grid on
```

The translation model that ode45 uses to simulate the golf ball is given below. Please note, the buoyancy force is calculated here so the same model could be used for both problem 9 and 10, but is always equal to 0 for the golf ball, due to setting the displaced volume of air equal to zero.

Translation Model

```
function [ dydt ] = translation_model( ~, y, p )
%translation_model Translation dynamics for a vehicle called by ode45
%   Kinematics and dynamics in inertial frame (N,E,D) coordinates
%   Inputs:
%       t       time, [s]
%       y       State
%       y = [x_E,       % North component of inertial position [m]
%            y_E,       % East component of inertial position [m]
%            z_E,       % Down component of inertial position [m]
%            xdot_E,    % North component of inertial velocity [m/s]
%            ydot_E,    % East component of inertial velocity [m/s]
%            zdot_E]    % Down component of inertial velocity [m/s]
%       p       parameters
%   Outputs:
%       dydt    state rate of change

    % Uses F_E (inertial coordinates) throughout
    m = p(1);
    g = p(2);
    rho = p(3);
    vol = p(4);
    cd = p(5);
    S = p(6);
    W = p(7:9);

    % Compute forces at the given time in F_E coordinates
    f_g = m*g*[0; 0; 1]; % Gravitational force [N]
    f_b = -vol*(rho)*g*[0; 0; 1]; % buoyancy force [N]

    % Relative wind (motion of the body relative to the air) in F_E
    % coordinates
    vrel = y(4:6) - W';

    vmag = norm(vrel);
    if vmag == 0
        vmag = 1;
    end

    f_d = -.5*rho*cd*S*dot(vrel,vrel)*vrel/vmag; % -0.5*rho*cd*S*sqrt(v(1)^2 + v(2)^2 +
v(3)^2)*v; % drag force [N]
    f = f_g + f_b + f_d; % total force

    % State derivatives
    dydt = [y(4); y(5); y(6); f(1)/m; f(2)/m; f(3)/m];
end
```

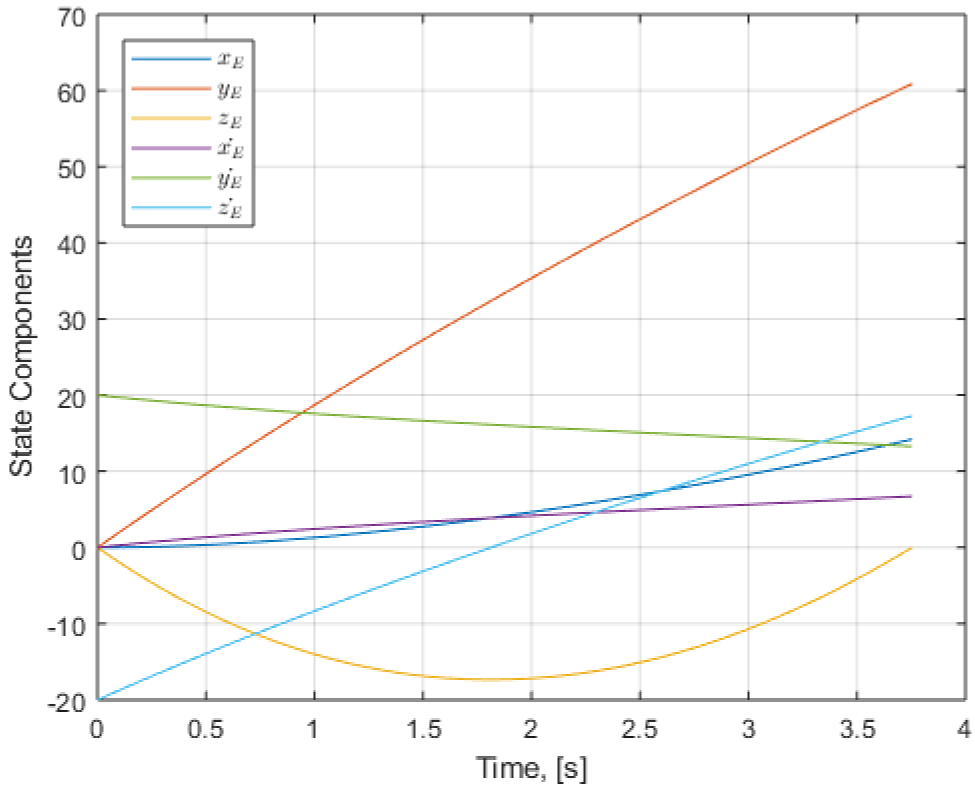


Figure 1: The components of the state vector with respect to time

Figure 1 gives the 6 components of the inertial state vector for the golf ball, $\boldsymbol{x} = [\bar{r}_E, \bar{V}_E^E]^T$, for the case of a 20 m/s North wind. The x_E coordinate starts from 0 and increases over time due to the North wind. The $\dot{x}_E$ component also increases as the wind accelerates it in that direction. The $\dot{y}_E$ component starts at the prescribed 20 m/s initial value, and decreases as the drag on the golf ball slows it down. The y_E coordinate starts from the initial value of 0 and increases over time, since the $\dot{y}_E$ values remains positive, which meets our expectation of the behavior of a golf ball hit in that direction. The z_E coordinate starts from 0, with an initial velocity $\dot{z}_E$ of -20 m/s, as prescribed, causing z_E to become negative (upward motion), reaching a maximum as gravity pulls the ball downward. The simulation was run until the upward motion returned to landing at z_E = 0. The simulation therefore behaves as expected for a ballistic object subject to drag.

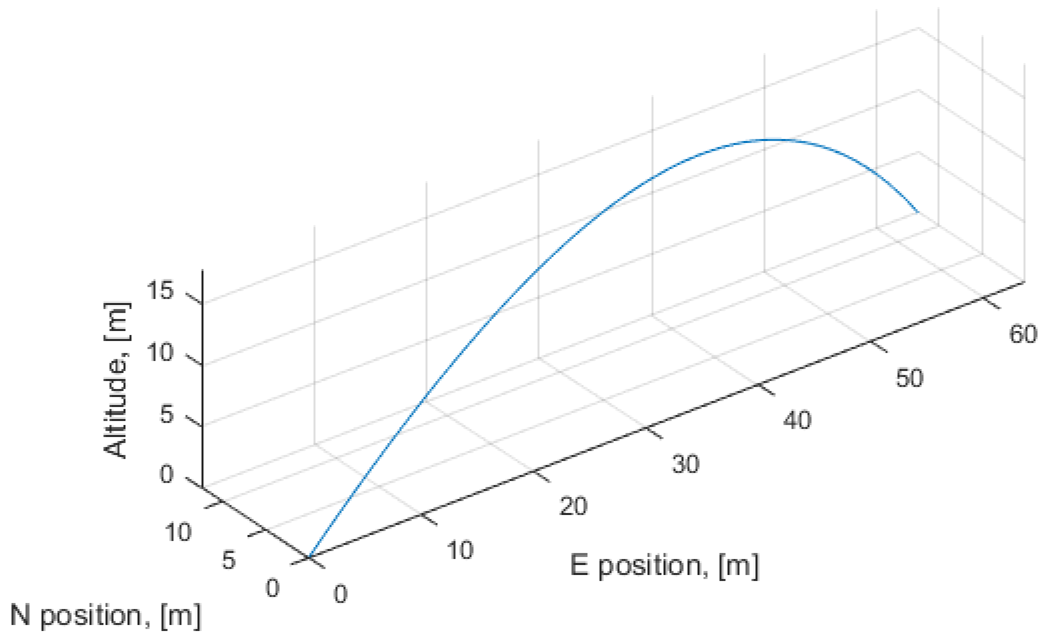


Figure 2: The trajectory of the golf ball. The negative of the D-component is plotted as altitude

Figure 2 shows the trajectory in 3D, using altitude as the negative of the Down component. Again, this makes sense given a Easterly and upward initial velocity, and a North wind.

9) (cont.) How sensitive is the landing location (characterized by a vertical displacement relative to the origin of zero) to horizontal wind, in m of deflection per m/s of wind?

How sensitive is the landing location?

```
crossWind = 1:1:20;

displacement = zeros(size(crossWind));
totalDisp = zeros(size(crossWind));

for i = 1:length(crossWind)
    w = [crossWind(i), 0, 0];

    p = [m, g, rho, vol, cd, S, w];

    % Simulate with ode45
    [t,y] = ode45(@(t,y) translation_model(t, y, p), tspan, IC, options);

    displacement(i) = norm(y(end, 1));
    totalDisp(i) = norm(y(end, 1:3));
```

```
end

figure()
plot(crossWind, totalDisp, 'x')
xlabel('North Wind Velocity [m/s]')
ylabel('Total Displacement from Origin [m]')
title('Effect of Wind on Total Displacement')
grid on

figure()
plot(crossWind, displacement, 'x')
xlabel('North Wind Velocity [m/s]')
ylabel('North Component of Displacement [m]')
title('Wind Sensitivity')
grid on
```

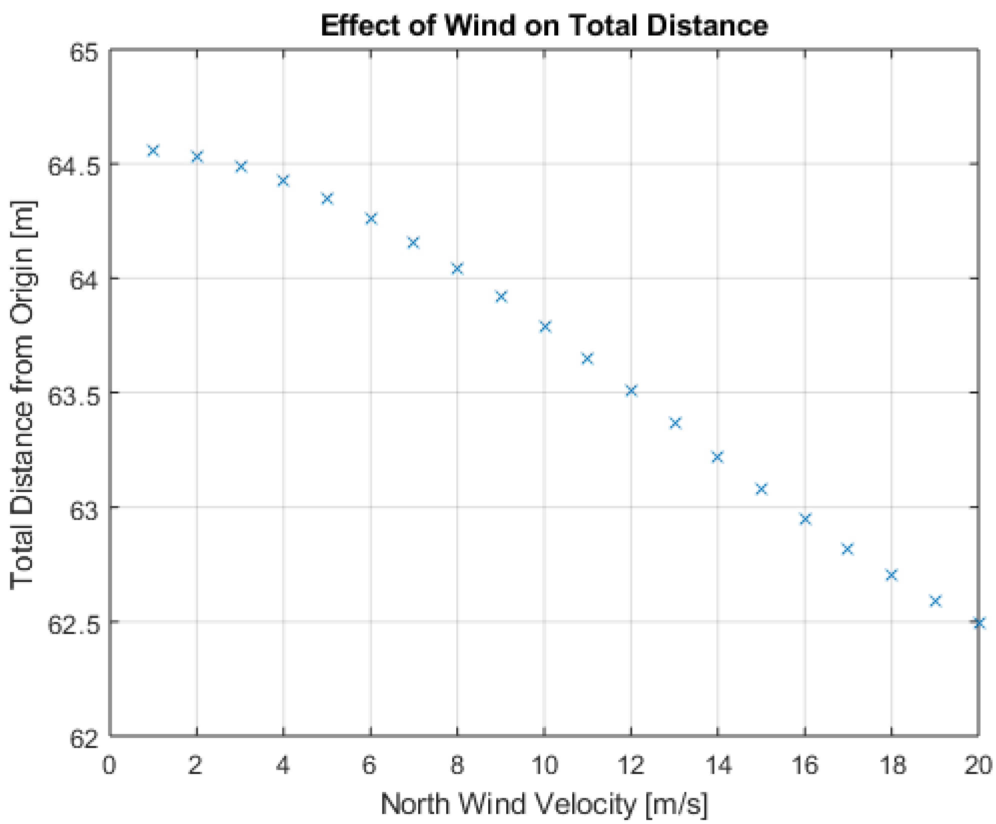


Figure 3: Magnitude of total displacement from the origin for several wind velocities

Figure 3 gives the results of the wind sensitivity analysis. The decrease in the magnitude of the total displacement from the origin can be explained by the increase in the magnitude of the drag force as the wind speed increases. The magnitude of the force of drag depends on the magnitude of the relative wind vector. So, an increase in one component of the relative wind vector will lead to an increase in the magnitude of the relative wind vector, causing some increase in all of the components of the force of drag, relative to the case of no wind.

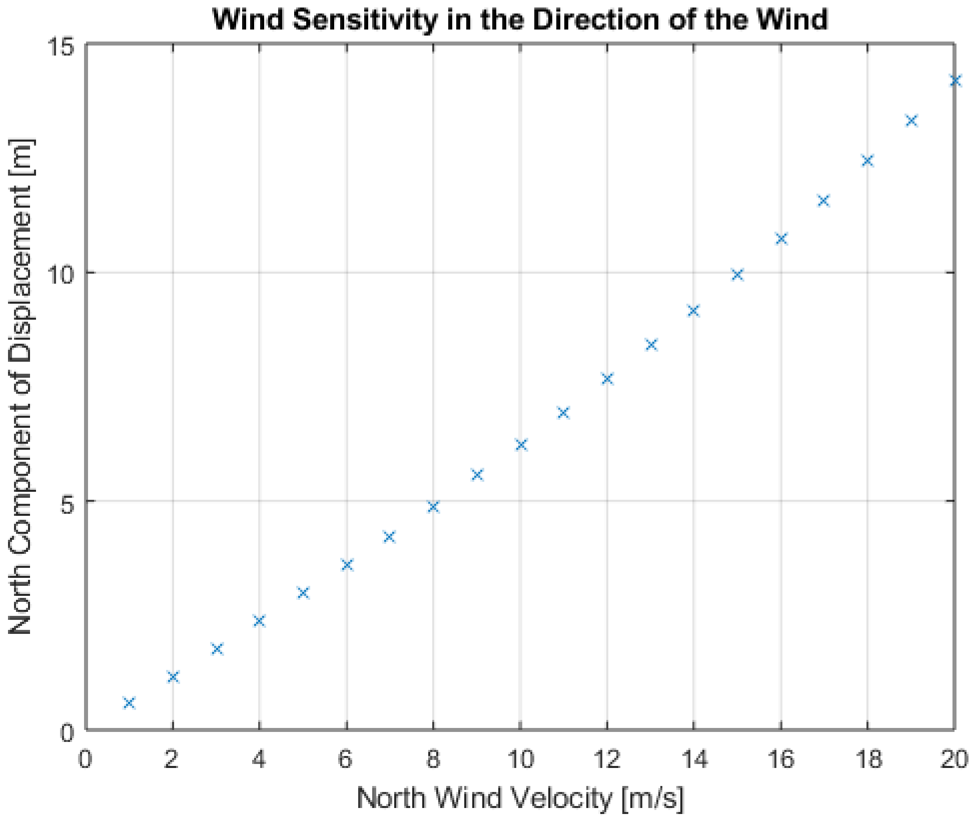


Figure 4: The sensitivity of the displacement in the direction of the wind.

Figure 4 shows the deviation of the landing location in the North direction, as a function of the speed of the North wind. As wind speed increases, the landing location displacement in that direction also increases, as expected. It would also be expected that the landing location would be much more sensitive in the North direction than the East direction due to a North wind, and this is also evident in the plots.

9) (cont.) If the initial velocity is constrained by a limited kinetic energy (i.e. due to human swing-strength limitations) equal to the case examined here, would longer distance be achieved by using a heavier or lighter golf ball?

Limited by kinetic energy

```
W = [0, 0, 0]; % Set wind equal to zero

v0 = [0; 20; -20];
m0 = 0.05;

mass = .001:.001:.1;

dist = zeros(size(mass));

for k = 1:length(mass)
    % Assuming the shot is in the same direction as before
    v = v0 * sqrt(m0/mass(k));
```

```
    IC = [0; 0; 0; v];
    p = [mass(k), g, rho, vol, cd, S, W];
    [t,y] = ode45(@(t,y) translation_model(t, y, p), tspan, IC, options);

    dist(k) = y(end, 2);
end

figure()
plot(mass, dist, 'x')
xlabel('Mass of Golf Ball [kg]')
ylabel('Distance at landing [m]')
title('Mass Sensitivity')
grid on
```

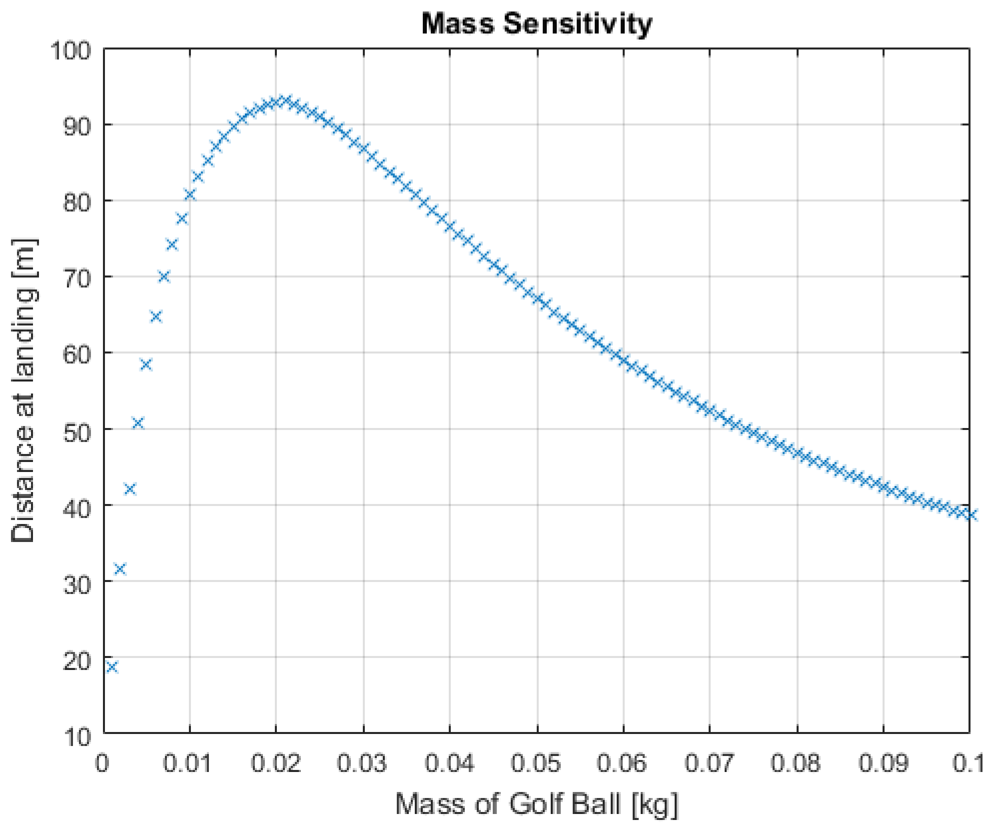


Figure 5: Distance from the origin versus mass of the golf ball

The initial kinetic energy for the case of a 0.05 kg ball and an initial velocity of 28.3 m/s is preserved as the ball mass m varies, if the initial velocity v satisfies $\frac{v}{v_o} = \sqrt{m_o/m}$. Figure 5 shows the resulting distance to landing (with zero wind) as the ball mass varies. For small golf ball mass, the flight distance is small because although the ball has a large initial velocity, it has a small momentum (like a whiffle ball). The large drag rapidly reduces the velocity. At large masses, the drag becomes less important because of the larger launch momentum, but the ball is brought down sooner due to gravity due to the lower initial velocity.

10) Repeat the simulation above, but also include buoyancy forces. Use this to model a balloon sonde, with balloon and sonde mass of 0.5kg. Simulate its behavior with 1 m^2 of He lifting gas from a ground release, displacing 1 m^2 of air, with coefficient of drag of 0.5. Assume a prevailing wind of 4 m/s North, 2 m/s East, and 0 m/s Down.

We use the same translation_model.m function as problem 9, but with a non-zero displaced volume of air to determine the buoyancy force and the weight of the lifting gas contributing to the overall mass, hence to the opposing gravity force.

```
% ASEN 3128 - Homework 1, Problem 10
%
% Translational dynamics simulation for an aircraft
% Attitude-independent drag, gravity, buoyancy forces
clc; clear; close all;

% Parameters
g = 9.81; % acceleration of gravity [m/s^2]
rho = 1.225; % air density [kg/m^3]

% Ode45 Set Up
options = odeset('Events',@ground_event);
```

Simulate

```
% Balloon Sonde parameters
m = 0.5; % mass [kg]
rho_lift = .179; % density of lifting gas [kg/m^3]
vol = 1.0; % volume displaced by lifting gas [m^3]
cd = 0.5; % coefficient of drag
r = (4*vol/(3*pi))^(1/3); % assume circular sonde
S = r^2*pi; % Planform area [m^2]
IC = [0; 0; 0; 0; 0; 0]; % Initial conditions
tspan = [0 120];

% Wind (direction air is moving relative to the inertial frame)
W = [4, 2, 0]; % wind velocity [North, East, Down] [m/s]

% State: y =
%   [x_E,       % North component of inertial position [m]
%    y_E,       % East component of inertial position [m]
%    z_E,       % Down component of inertial position [m]
%    xdot_E,    % North component of inertial velocity [m/s]
%    ydot_E,    % East component of inertial velocity [m/s]
%    zdot_E]    % Down component of inertial velocity [m/s]
p = [m, g, rho, vol, cd, S, W];
```

```
% Simulate with ode45
[T,Y] = ode45(@(t,y) translation_model(t, y, p), tspan, IC, options);

figure()
plot(T, Y)
xlabel('Time, [s]')
ylabel('State Components')
legend({'$x_E$', '$y_E$', '$z_E$', '$\dot{x_E}$', '$\dot{y_E}$', '$\dot{z_E}$'},...
    'Interpreter', 'latex', 'Location', 'NorthWest')

figure()
plot3(Y(:,2), Y(:,1), -Y(:,3))
xlabel('E position, [m]')
ylabel('N position, [m]')
zlabel('Altitude, [m]')
axis equal
grid on
```

Translation Model

```
function [ dydt ] = translation_model( ~, y, p )
%translation_model Translation dynamics for a vehicle called by ode45
%   Kinematics and dynamics in inertial frame (N,E,D) coordinates
%   Inputs:
%       t       time, [s]
%       y       State
%       y = [x_E,       % North component of inertial position [m]
%            y_E,       % East component of inertial position [m]
%            z_E,       % Down component of inertial position [m]
%            xdot_E,    % North component of inertial velocity [m/s]
%            ydot_E,    % East component of inertial velocity [m/s]
%            zdot_E]    % Down component of inertial velocity [m/s]
%       p       parameters
%   Outputs:
%       dydt    state rate of change

    % Uses F_E (inertial coordinates) throughout
    m = p(1);
    g = p(2);
    rho = p(3);
    vol = p(4);
    cd = p(5);
    S = p(6);
    W = p(7:9);

    % Compute forces at the given time in F_E coordinates
    f_g = m*g*[0; 0; 1]; % Gravitational force [N]
    f_b = -vol*(rho)*g*[0; 0; 1]; % buoyancy force [N]

    % Relative wind (motion of the body relative to the air) in F_E
    % coordinates
    vrel = y(4:6) - W';
```

```
    vmag = norm(vrel);
    if vmag == 0
        vmag = 1;
    end

    f_d = -.5*rho*cd*S*dot(vrel,vrel)*vrel/vmag; % -0.5*rho*cd*S*sqrt(v(1)^2 + v(2)^2 +
v(3)^2)*v; % drag force [N]
    f = f_g + f_b + f_d; % total force

    % State derivatives
    dydt = [y(4); y(5); y(6); f(1)/m; f(2)/m; f(3)/m];
end
```

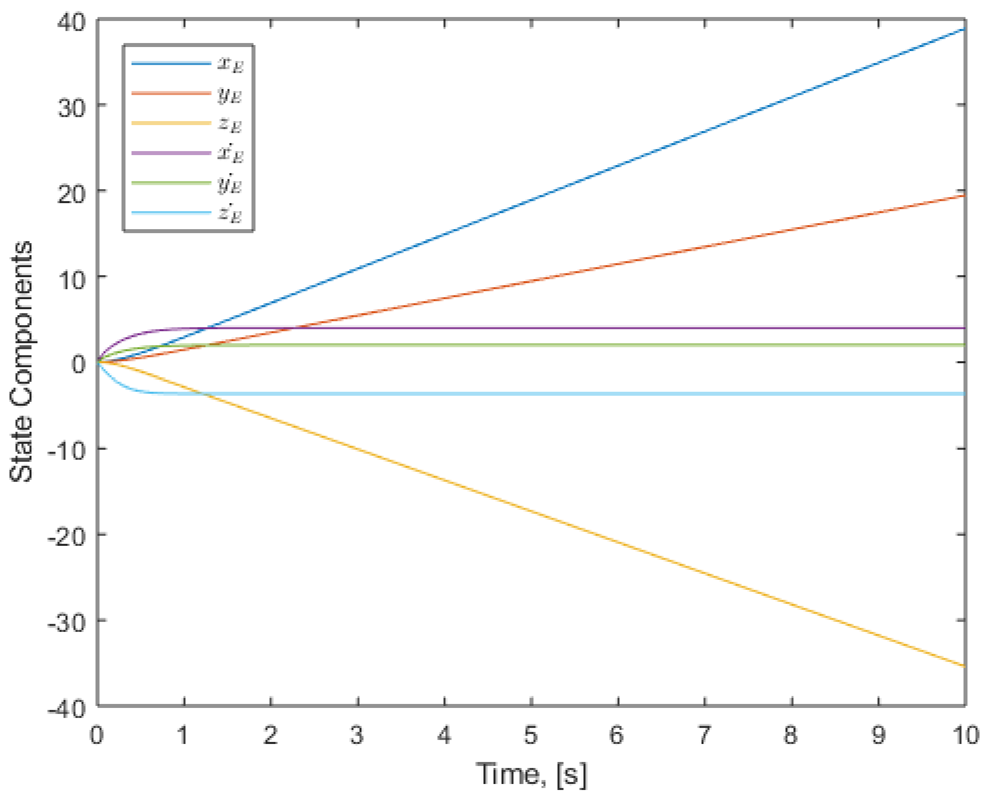


Figure 6: Components of the balloon's state vector versus time

The forces of gravity and buoyancy are assumed to be constant here, so the only variable force is drag which depends on the relative wind vector. Figure 6 gives the simulated components of the balloon's state vector, $\boldsymbol{x} = [\bar{r}_E, \bar{V}_E^E]^T$ after balloon release. Note the inertial velocity, $[\dot{x}_E, \dot{y}_E, \dot{z}_E]^T$ reaches terminal velocity within the first second of flight. From this point on, the net force on the balloon is equal to 0 in all three directions. With 0 acceleration, the velocity is constant, and the inertial position components increase linearly in time.

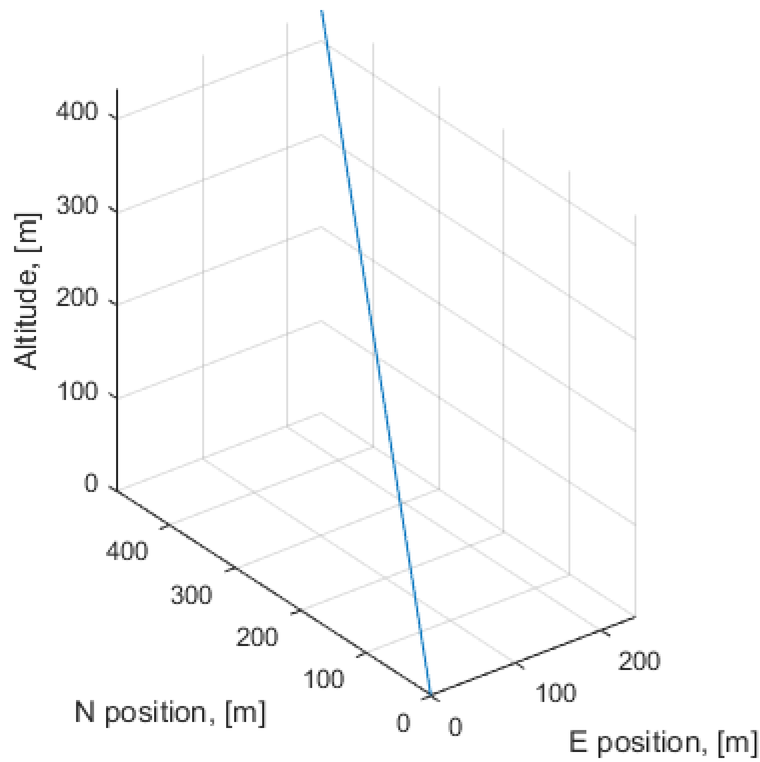


Figure 7: Graphical representation of the balloon's motion. The negative of the D-component is plotted as altitude.

Figure 7 shows the 3D trajectory. The buoyancy force exceeds gravity, causing the balloon to develop an upward velocity, and the drag force will cause the balloon to travel in the direction of the wind (toward the North in this case).

10(cont) Construct a relationship between wind speed (from 0 to 20 m/s) and required He volume to maintain an accent angle from vertical of less than 45 degrees.

The code below sets a wind speed, then increases the balloon radius until a 45 degree ascent angle is obtained.

Relationship between wind speed and required volume

```
windSpeed = 0.5:.5:20;
volReq = zeros(size(windSpeed));
rAscent = zeros(size(windSpeed));

wHat = W/norm(W);
m0 = .5;

% Reduce runtime
tspan = [0 20];

for i = 1:length(windSpeed)
```

```
    % To fly (f_b = f_g):
    r0 = (3*m/(4*pi*(rho-rho_lift)))^(1/3); %(4*(m0/(rho-rho_lift))/(3*pi))^(1/3);
    rstep = windSpeed(i)/500;
    r = r0;% + rstep;
    gamma0 = 90;

    W = [windSpeed(i), 0, 0];% * wHat;

    % look for the 45 degree ascent
    for k = 1:100      % counter
        % total system mass
        m = 0.5 + rho_lift*(4/3)*pi*r^3;

        vol = 4/3*pi*r^3;
        S = r^2*pi; % Planform area [m^2]

        p = [m, g, rho, vol, cd, S, W];
        % Simulate with ode45
        [t,y] = ode45(@(t,y) translation_model(t, y, p), tspan, IC, options);

        % calculate angle
%         vert = -y(end,3);
%         lat = norm(y(end,1:2));
        vert = -y(end,3);
        lat = norm(y(end,1:2));
        gamma = atand(lat/vert);

        % Find r if the ascent angle is greater than 45
        if gamma < 45
            rAscent(i) = interp1([gamma0 gamma],[r0 r],45);
            volReq(i) = (4/3)*pi*rAscent(i)^3;
            fprintf('Found volume necessary for 45 degree ascent at i = %d\n', i)
            break;
        else
            gamma0 = gamma;
            r0 = r;
            rstep = rstep + k/10000;
            r = r + rstep;
        end
    end
end

% plot the results
figure
semilogy(windSpeed, volReq, 'x')
xlabel('Wind Speed [m/s]')
ylabel('Helium Volume [m^3]')
title('Balloon Volume Requirements')
grid on
```

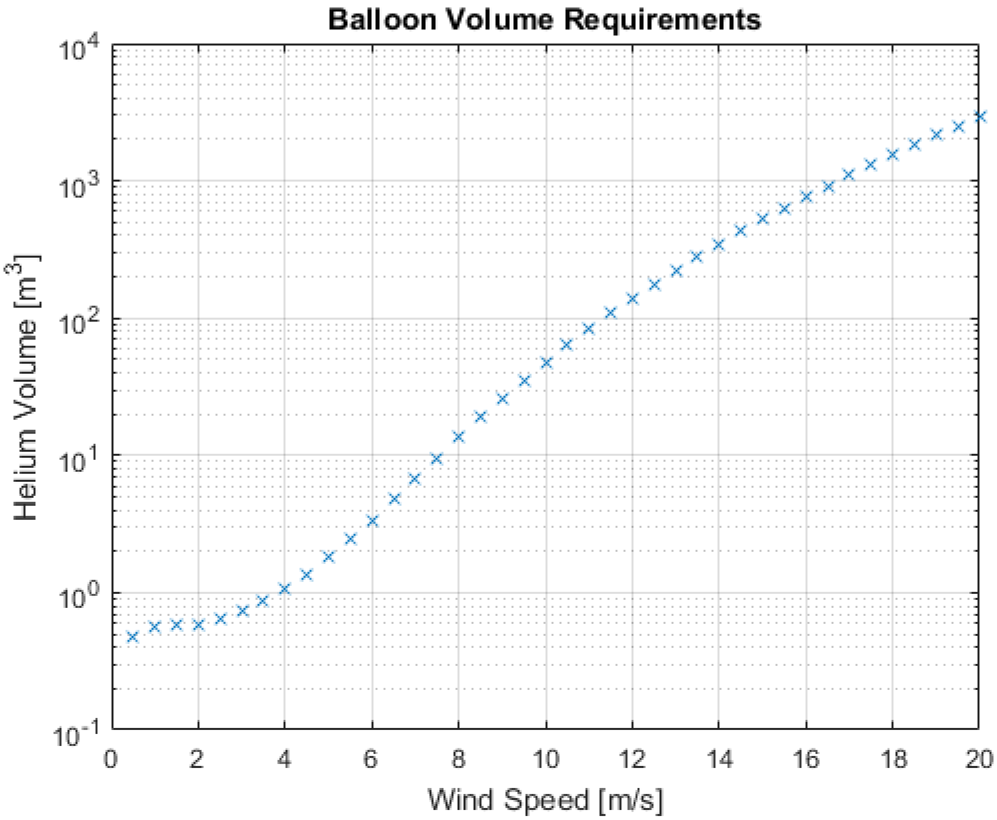


Figure 8: The volume necessary to achieve a 45° ascent angle at various wind speeds. Volumes above the given line will have a smaller ascent angle from vertical.

Figure 8 provides the resulting relationship between the balloon volume and wind speed. The balloon's initial radius was set so that the buoyancy force equaled the force of gravity, so with any wind, the initial ascent angle from vertical was 90°, exceeding the requirement, so a larger volume is needed. At lower wind speeds, only small changes in the balloon's volume are needed to achieve the 45° ascent angle because the lateral force due to the wind was so small. As the wind speed (and drag) increases, the volume required for the 45° ascent angle increased significantly. As described in problem 9, all the components of the drag force vector increase when relative wind increases. Additionally, the increased volume of Helium means that the planform area of the balloon also increases, further increasing the force of drag. So, the force necessary to overcome drag and gain altitude scales non-linearly with wind speed in Figure 8.