Cairo University
Faculty of Engineering
Aerospace Department

AER 408

Task 1

Submitted to: Prof. Osama Saaid
Eng .Wessam

Team 4

Team members

name	sec	bn
Mohammed Ahmed Hassan Ahmed	**2**	**37**
Ibrahim Thabet Allam	**1**	**1**
Mohammed Hatem mohammed Saeed	**2**	**39**
Mohamed Hassan Gad Ali	**2**	**41**
Mohammed Abdelmawgood Ghoneam	**2**	**43**

Autopilot literature review

Research questions

a) What is an autopilot?

An autopilot is a system that allows an airplane, marine vehicle, or spaceship to navigate without the need for constant manual control by a human operator. Human operators are not replaced by autopilots. Instead, the autopilot supports the driver in maintaining vehicle control.

b) What are the inputs & outputs of Autopilot system onboard an airplane?

Inputs:

Control Inputs:

1. Radio Control Receivers
2. MAVLink Data Streams, ie ground control stations or companion computers

Sensor inputs:

GPS, Compass, Airspeed, Rangefinders, IMU

Power Management Unit Inputs:

1. Received Signal Strength Input (RSSI)
2. Analog Airspeed Sensors

Outputs:

1. ESCs (electronic circuit that controls and regulates the speed of an electric motor) for motors
2. Servos for control surfaces

3. Telemetry data
4. Actuators and General Purpose I/O like LEDs, buzzers etc

c) What would be the role of the pilot in an airplane equipped with an autopilot?

The pilot sets the flight plan and turn on the autopilot sometimes the pilot reprograms the autopilot in case if worked incorrectly. autopilot is not smart enough to fly a plane by It self

d) What is the difference between Autopilot & SAS?

SAS (Stability augmentation system) generally used during low and slow maneuvering where the pilot may be making constant attitude changes in preparation for landing.

Autopilot do same functions as SAS in addition it provides more functions ,Autopilot Is more sophisticated than SAS

e) What is the role of the onboard sensors like (GPS, gyroscopes, ...etc.)? give example.

Sensors provide the autopilot computers with data like speed, coordinates, position so the computers can estimate the states and give the correct control actions to the actuators for example

Gyroscope an IMU supplies the autopilot with position data so if the airplane in incorrect position it gives the actuator signal so that it can adjust position

Flight Mechanics review

a) Assume a mass element at a rigid body, the velocity of this mass w.r.t. fixed axes:

$$\vec{V} = \overrightarrow{V_c} + \frac{d\vec{\boldsymbol{r}}}{dt} \qquad (1)$$

By using the calculations of linear momentum:

The force at this point is given by:

$$\delta\vec{F} = \delta m \frac{d\vec{V}}{dt} \qquad (2)$$

Sub. By (1) in (2) and take summation for all point masses is in the rigid body:

$$\vec{F} = \sum \delta\vec{F} = \frac{d}{dt}\sum(\overrightarrow{V_c} + \frac{d\vec{\boldsymbol{r}}}{dt})\delta m$$

$$\vec{F} = m\frac{d\overrightarrow{V_c}}{dt} + \frac{d}{dt}\sum \frac{d\vec{\boldsymbol{r}}}{dt}\,\delta m = m\frac{d\overrightarrow{V_c}}{dt} + \frac{d^2}{dt^2}\sum r\,\delta m$$

Since: $r\delta m = 0$

$$\therefore \vec{F} = m\frac{d\overrightarrow{V_c}}{dt}$$

From general definition:

$$\frac{d\vec{A}}{dt}_I = \frac{d\vec{A}}{dt}_{\boldsymbol{B}} + \omega \times \vec{A}$$

So:

$$\vec{F} = m\frac{d\overrightarrow{V_c}}{dt} + m(\omega \times \overrightarrow{V_c})$$

Where:

$$\vec{F} = F_x\vec{\imath} + F_y\vec{\jmath} + F_z\vec{k}, \qquad \vec{\omega} = P\vec{\imath} + q\vec{\jmath} + r\vec{k}, \qquad \overrightarrow{V_c} = u\boldsymbol{i} + v\boldsymbol{j} + w\boldsymbol{k}$$

So:

$$F_x = m(\dot{u} + qw - rv)$$

$$F_y = m(\dot{v} + ru - pw)$$

$$F_z = m(\dot{w} + pv - qu)$$

By using the calculations of angular momentum:

$$\delta\vec{M} = \frac{d}{dt}\delta\vec{H} = \frac{d}{dt}(\vec{r} \times \vec{V})\delta m$$

By using the upper general definition: 0

$$\frac{d\vec{r}}{dt}_I = \frac{d\vec{r}}{dt}_{\boldsymbol{B}} + \omega \times \vec{r}$$

$$Then: \qquad \vec{H} = \sum \delta\vec{H} = \sum(\vec{r} \times \overrightarrow{V_c})\delta m + \sum[r \times (\omega \times \vec{r})]\,\delta m$$

Where: $\vec{r} = x\boldsymbol{i} + y\boldsymbol{j} + z\boldsymbol{k}$

Then:

$$H_x = p\sum(y^2 + z^2)\,\delta m - q\sum xy\,\delta m - r\sum xz\,\delta m$$

$$H_y = -p\sum xy\,\delta m + q\sum(x^2 + z)^2\,\delta m - r\sum yz\,\delta m$$

$$H_z = -p\sum xz\,\delta m + q\sum yz\delta m - r\sum(x^2 + y^2)\;\delta m$$

By expressing the above equations in terms of moment of inertia about body axes:

$$H_x = pI_x - qI_{xy} - rI_{xz}$$

$$H_y = -pI_{xy} + qI_y - rI_{yz}$$

$$H_z = -pI_{xz} - qI_{yz} + rI_z$$

Since:

$$\vec{M} = \frac{d}{dt}\vec{H} = \frac{d\vec{H}}{dt}_B + \vec{\omega} \times \vec{H}$$

$$L = \dot{H}_x + qH_z - rH_y$$

$$M = \dot{H}_y + rH_x - pH_z$$

$$N = \dot{H}_z + pH_y - qH_x$$

b)

i. Kinetics equations

$$F_x = m(\dot{u} + qw - rv)$$

$$F_y = m(\dot{v} + ru - pw)$$

$$F_z = m(\dot{w} + pv - qu)$$

$$L = \dot{H}_x + qH_z - rH_y$$

$$M = \dot{H}_y + rH_x - pH_z$$

$$N = \dot{H}_z + pH_y - qH_x$$

ii. Kinematics equations:

$$\vec{V} = \vec{V_c} + \frac{d\vec{\boldsymbol{r}}}{dt}$$

$$H_x = pI_x - qI_{xy} - rI_{xz}$$

$$H_y = -pI_{xy} + qI_y - rI_{yz}$$

$$H_z = -pI_{xz} - qI_{yz} + rI_z$$

c) The assumptions are:

i. The body is rigid which means the mass is constant and there is no change in geometry (Stress calculations are neglected).

ii. Fix a body axes at the center of mass of the rigid body.

d) Now we are dealing with an aircraft which means that $I_{xy} = I_{yz} = 0$

$(symmetry\ cond.)$

Then: The airplane translational and rotational EOM are:

$$F_x = m(\dot{u} + qw - rv)$$

$$F_y = m(\dot{v} + ru - pw)$$

$$F_z = m(\dot{w} + pv - qu)$$

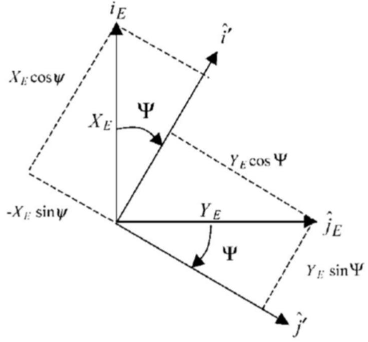
$$L = I_x\dot{p} - I_{xz}\dot{r} + qr(I_z - I_y) - I_{xz}pq$$

$$M = y\dot{q} + rp(I_x - I_z) + I_{xz}(p^2 - r^2)$$

$$N = -I_{xz}\dot{p} + I_z\dot{r} + pq(I_y - I_x) + I_{xz}qr$$

By adding the calculations of earth axis to body axis transformation:

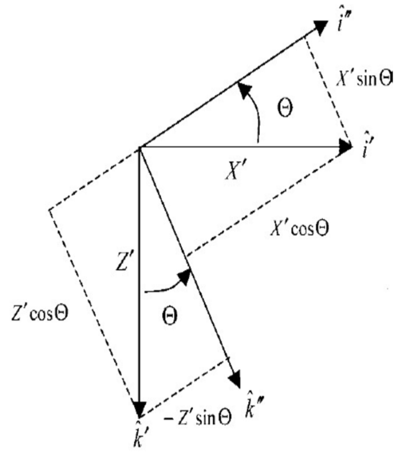
$$\begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \end{bmatrix} = \begin{bmatrix} cos\psi & sin\psi & 0 \\ -sin\psi & cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_E \\ Y_E \\ Z_E \end{bmatrix}$$



$$\boldsymbol{F_1} = R_3(\psi)\boldsymbol{F_E}$$

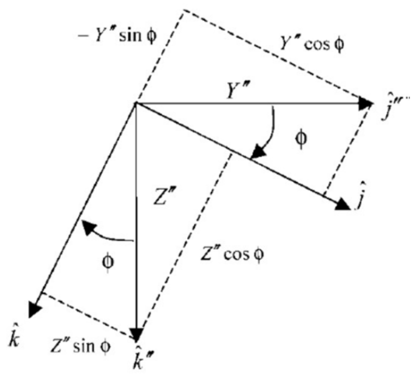
$$\begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \end{bmatrix} = \begin{bmatrix} cos\theta & 0 & -sin\theta \\ 0 & 1 & 0 \\ sin\theta & 0 & cos\theta \end{bmatrix} \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \end{bmatrix}$$

$$\boldsymbol{F_2} = R_2(\theta)\boldsymbol{F_1} = R_2(\theta)R_3(\psi)\boldsymbol{F_E}$$



$$\begin{bmatrix} X_B \\ Y_B \\ Z_B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\phi & sin\phi \\ 0 & -sin\phi & cos\phi \end{bmatrix} \begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \end{bmatrix}$$

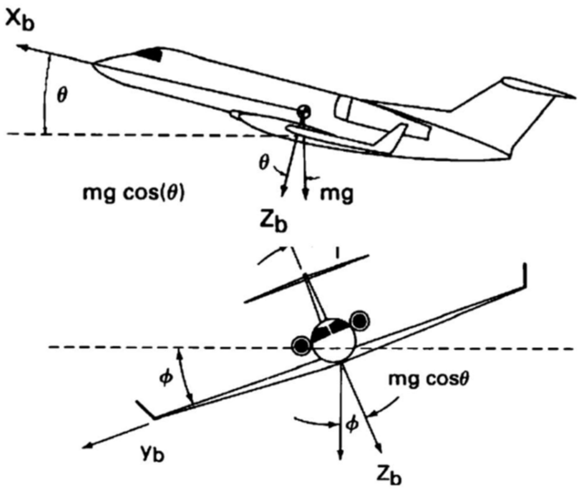
$$\boldsymbol{F_B} = R_1(\phi)\boldsymbol{F_2} = R_1(\phi)R_2(\theta)R_3(\psi)\boldsymbol{F_E}$$



Let’s apply the above transformation on the gravitational force of the airplane as following:

$$\vec{F}_{Gravity_B} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\phi & sin\phi \\ 0 & -sin\phi & cos\phi \end{bmatrix} \begin{bmatrix} cos\theta & 0 & -sin\theta \\ 0 & 1 & 0 \\ sin\theta & 0 & cos\theta \end{bmatrix} \begin{bmatrix} cos\psi & sin\psi & 0 \\ -sin\psi & cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix}_E$$

$$\vec{F}_{Gravity_B} = \begin{bmatrix} -mgsin\theta \\ mgsin\phi cos\theta \\ mgsin\theta cos\phi \end{bmatrix}_B$$



Then:

$$-mgsin\theta + F_{T_X} = m(\dot{u} + qw - rv)$$

$$mgsin\phi cos\theta + F_{T_Y} = m(\dot{v} + ru - pw)$$

$$mgsin\theta cos\phi + F_{T_Z} = m(\dot{w} + pv - qu)$$

$$\begin{Bmatrix} P \\ Q \\ R \end{Bmatrix} = \begin{bmatrix} 1 & 0 & -sin\theta \\ 0 & cos\phi & cos\theta sin\phi \\ 0 & -sin\phi & cos\phi cos\theta \end{bmatrix} \begin{Bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{Bmatrix}$$

$$\begin{Bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{Bmatrix}_E = \begin{bmatrix} C_\theta C_\psi & S_\phi S_\theta C_\psi - C_\phi S_\psi & C_\phi S_\theta C_\psi + S_\phi S_\psi \\ C_\theta S_\psi & S_\phi S_\theta S_\psi + C_\phi C_\psi & C_\phi S_\theta S_\psi - S_\phi C_\psi \\ -S_\theta & S_\theta C_\theta & C_\phi C_\theta \end{bmatrix} \begin{Bmatrix} u \\ v \\ w \end{Bmatrix}_B$$

c) The EOM's are coupled, nonlinear and first order DE.

f)

i. Body axes: they are a set of axes which are fixed at the body in its translational and rotations.

ii. Inertial axes: they are a set of axes which are fixed at a specified position on the ground.

g)

i. **Pitch angle**: it's formed due to rotation about y-axis.

ii. **AOA**: it's formed due to due to a difference between the direction of flow and the plane wing.

iii. **Sideslip angle**: it's formed due to due to lateral deviation between the plane and the direction of the flow.

iv. **Yaw angle (heading)**: it's formed due to due to the rotation about z-axis.

Numerical solutions of ODEs

Some numerical solving algorithms for ODE

one step methods, are methods that involve only y_i and intermediate quantities to compute the next value y_{i+1}. In contrast, multi-step methods use more than the previous point

One-step methods, like:

Euler-Cauchy method

Improved Euler method

Raunge-Kutta method

Backward Euler method

Multistep methods, like:

Adams-Bash forth method

Adams-Moulton method

Other methods

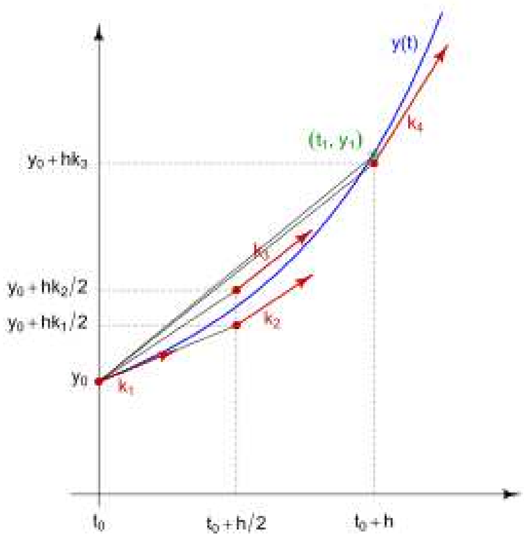
Predictor-Corrector Methods

Exponential integrator methods

The chosen method for solving the aircraft EOM.

We choose to use Raunge-Kutta method since its one of the most accurate one step methods used in the numerical solutions, in addition it's the method used in the MATLAB ODE45 algorithm.

The order of the Raunge-Kutta indicates the number of slopes used in the approximation; so higher orders mean higher accuracy of the solution; the wights of the slopes is determined comparing the coefficients with Taylor expansion of the function.[1] anyway, the fourth order Raunge-Kutta is commonly as it gives good accuracy and acceptable computational power.



Initial conditions:

$$u_0, v_0, w_0, x_0, y_0, z_0, \theta_0, \varphi_0, \psi_0, p_0, q_0, r_0$$

Inputs:

$$\delta_a, \delta_e, \delta_r, \delta_T$$

Outputs:

Components of air craft velocity: $u\ , v\ , w$
Position: $x\ , y\ , z$
Euler angles: $\theta\ , \varphi\ , \psi$
Angular velocity components: $p\ , q\ , r$

[1] for more details regarding the derivation of Raunge-Kutta method use the following link https://www.math.hkust.edu.hk/~machas/numerical-methods-for-engineers.pdf?fbclid=IwAR1mOjc35AS3QV_ZuL0Z_nV8Ph3nCop4FlsaS_ueBMz31whmOBPh9LJ2GlM

The general solution of 4th order Runge-Kutta method

$$k_1 = \Delta t f(t_n, x_n)$$

$$k_2 = \Delta t f\left(t_n + \frac{1}{2}\Delta t, x_n + \frac{1}{2}k_1\right)$$

$$k_3 = \Delta t f\left(t_n + \frac{1}{2}\Delta t, x_n + \frac{1}{2}k_2\right)$$

$$k_4 = \Delta t f(t_n + \Delta t, x_n + k_3)$$

$$x_{n+1} = x_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

For the following system of ODEs

$$f = \frac{dy_1}{dt} = \sin(t) + \cos(y_1) + \sin(y_2)$$
$$g = \frac{dy_2}{dt} = \cos(t) + \sin(y_2)$$

IC $@t = 0,\quad y_1 = 0, y_2 = 0 \qquad t_f = 20\ sec \qquad n = 100 \qquad h = \frac{\Delta t}{n}$

$$K_1 = hf(t_n, y_n^1, y_n^2)$$

$$K_2 = hf\left(t_n + \frac{1}{2}h, y_n^1 + \frac{1}{2}K_1, y_n^2 + \frac{1}{2}q_1\right)$$

$$K_3 = hf\left(t_n + \frac{1}{2}h, y_n^1 + \frac{1}{2}K_2, y_n^2 + \frac{1}{2}q_2\right)$$

$$K_4 = hf(t_n + h, y_n^1 + K_3, y_n^2 + q_3)$$

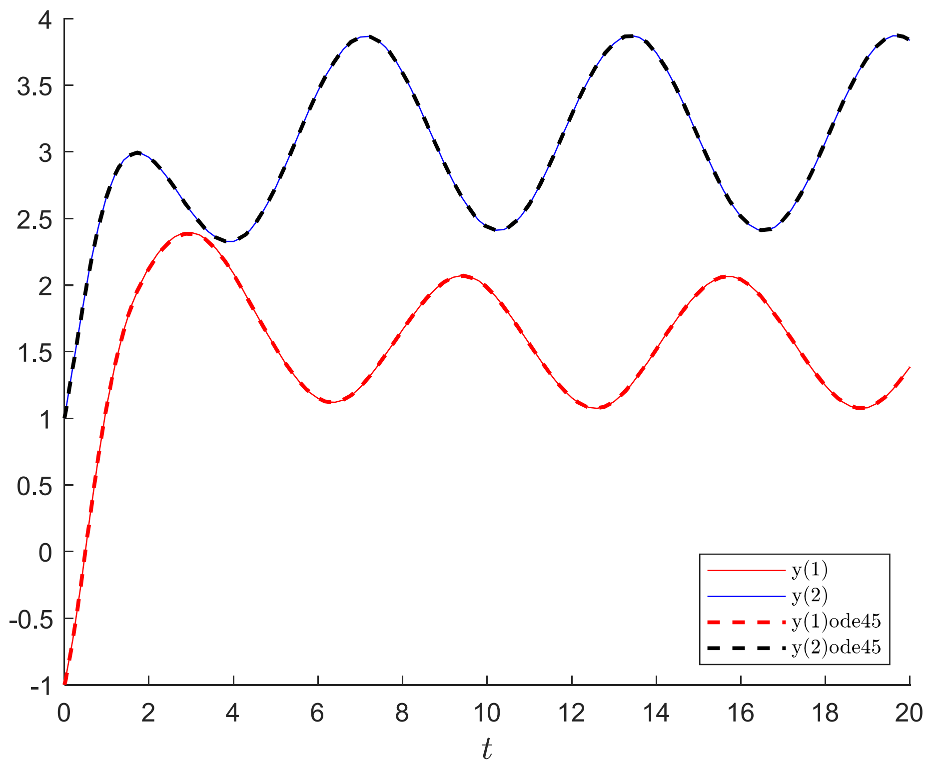
$$q_1 = hg(t_n, y_n^1, y_n^2)$$

$$q_2 = hg\left(t_n + \frac{1}{2}h, y_n^1 + \frac{1}{2}K_1, y_n^2 + \frac{1}{2}q_1\right)$$

$$q_3 = hg\left(t_n + \frac{1}{2}h, y_n^1 + \frac{1}{2}K_2, y_n^2 + \frac{1}{2}q_2\right)$$

$$q_4 = hg(t_n + h, y_n^1 + K_3, y_n^2 + q_3)$$

$$y_{n+1}^1 = y_n^1 + \frac{1}{6}(q_1 + 2q_2 + 2q_3 + q_4)$$
$$y_{n+1}^2 = y_n^2 + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$



MATLAB code

```
clc
clear All
close All

t1=0;
t2=20;
n=100;
h=(t2-t1)/n;
t=(linspace(t1,t2,n+1))';
y1=zeros(length(t),1);
y2=zeros(length(t),1);
y2(1)=1;
y1(1)=-1;


for n=1:length(t)-1
k1=h*(cos(t(n))+sin(y2(n)));
q1=h*(sin(t(n))+cos(y1(n))+sin(y2(n)));

k2=h*(cos(t(n)+0.5*h)+sin(y2(n)+0.5*k1));
q2=h*(sin(t(n)+0.5*h)+cos(y1(n)+0.5*q1)+sin(y2(n)+0.5*k1));

k3=h*(cos(t(n)+0.5*h)+sin(y2(n)+0.5*k2));
q3=h*(sin(t(n)+0.5*h)+cos(y1(n)+0.5*q2)+sin(y2(n)+0.5*k2));

k4=h*(cos(t(n)+h)+sin(y2(n)+k3));
q4=h*(sin(t(n)+h)+cos(y1(n)+q3)+sin(y2(n)+k3));

y2(n+1)=y2(n)+(1/6)*(k1+2*k2+2*k3+k4);
y1(n+1)=y1(n)+(1/6)*(q1+2*q2+2*q3+q4);
end


figure
hold on
plot(t,y1,'r')
plot(t,y2,'b')
legend({'y(1)','y(2)'},...
'Location','southeast','FontSize',8,'Interpreter','latex')
xlabel('$t$','Interpreter','latex','FontSize',13)
%% Check using ode45
[tv,Yv]=ode45(@sys_fun,[0 20],[-1 1]);
plot(tv,Yv(:,1),'r--','LineWidth',1.5)
```

```
plot(tv,Yv(:,2),'k--','LineWidth',1.5)
legend({'y(1)','y(2)','y(1)ode45','y(2)ode45'},...
'Location','southeast','FontSize',8,'Interpreter','latex')
xlabel('$t$','Interpreter','latex','FontSize',13)
function f=sys_fun(t,Y)
f(1,1)=sin(t)+cos(Y(1))+sin(Y(2));
f(2,1)=cos(t)+sin(Y(2));
end
```