Contents


```
clear all; close all; clc;

Ixx = 1.82*10^7; %slugs-ft^2
Izz = 4.97*10^7; %slugs-ft^2
Ixz = 9.70*10^5; %slugs-ft^2
S = 5500; %ft^2
b = 195.7; %ft
u1 = 275; %ft/s Mach 0.25 at Sea Level
theta = 2.4/180*pi; % rad Arbitrary
rho = 0.002378; % slugs/ft^3
q = 1/2*rho*u1^2;
g = 32.17405; % ft/s^2
m= 636636/g; % lb
```

Dimensional stability Derivatives

```
Cyb=-0.96;
Clb=-0.221;
Cnb=0.15;
Clp=-0.45;
Cnp=-0.121;
Clr=0.101;
Cnr=-0.3;
Clda=0.0461;
Cnda=0.0064;
Cydr=0.175;
Cldr=0.007;
Cndr=-0.109;
Cyp = 0;
Cyr = 0;
Cyda = 0;
CnTb = 0;
CnTr = 0;


Yb = q*S/m*Cyb;
Yp = q*S*b/(2*m*u1)*Cyp;
Yr = q*S*b/(2*m*u1)*Cyr;
Yda = q*S/m*Cyda;
Ydr = q*S/m*Cydr;
Lb  = q*S*b/Ixx*Clb;
Lp = q*S*b^2/(2*Ixx*u1)*Clp;
Lr = q*S*b^2/(2*Ixx*u1)*Clr;
Lda = q*S*b/Ixx*Clda;
Ldr = q*S*b/Ixx*Cldr;
Nb  = q*S*b/Izz*Cnb;
NTb = q*S*b/Izz*CnTb;
Np = q*S*b^2/(2*Izz*u1)*Cnp;
Nr = q*S*b^2/(2*Izz*u1)*Cnr;
NTr = q*S*b^2/(2*Izz*u1)*CnTr;
```

```
Nda = q*S*b/Izz*Cnda;
Ndr = q*S*b/Izz*Cndr;
```

Set up state space equation

```
M = [u1 0 0 0 0;
    0 1 -Ixz/Ixx 0 0;
    0 -Ixz/Izz 1 0 0;
    0 0 0 1 0;
    0 0 0 0 1];

R= [Yb Yp Yr-u1 g*cos(theta) 0;
    Lb Lp Lr 0 0;
    Nb+NTb Np Nr+NTr 0 0;
    0 1 tan(theta) 0 0;
    0 0 sec(theta) 0 0];
F = [Yda Ydr;
    Lda Ldr;
    Nda Ndr;
    0 0;
    0 0];

A = inv(M)*R;
B = inv(M)*F;
C = eye(5);
D = zeros(5,2);
```

Problem 1 a)

```
[eVec,eVal] = eig(A);

% Roll
eValRoll = eVal(:,2);
wdRoll   = imag(eValRoll(2));
wnRoll   = sqrt((real(eValRoll(2)))^2+(wdRoll)^2);
dampRoll = abs(real(eValRoll(2))/wnRoll);
delTRoll = log(2)/abs(real(eValRoll(2)));
NRoll    = log(2)*wdRoll/abs(real(eValRoll(2))*2*pi);

fprintf ("Roll-----------------------------------------------------------")
fprintf ("\nDamped Frequency: %f\n", wdRoll)
fprintf ("Natural Frequency: %f\n", wnRoll)
fprintf("Damping ratio: %f\n", dampRoll)
fprintf("Time to damp to half the initial amplitude: %f\n", delTRoll)
fprintf("The number of cycles to damp to half the initial amplitude: %f\n",NRoll)

% Dutch Roll
eValDutch = eVal(:,3);
wdDutch   = imag(eValDutch(3));
wnDutch   = sqrt((real(eValDutch(3)))^2+(wdDutch)^2);
dampDutch = abs(real(eValDutch(3))/wnDutch);
delTDutch = log(2)/abs(real(eValDutch(3)));
NDutch    = log(2)*wdDutch/abs(real(eValDutch(3))*2*pi);

fprintf ("Dutch Roll-----------------------------------------------------")
fprintf ("\nDamped Frequency: %f\n", wdDutch)
fprintf ("Natural Frequency: %f\n", wnDutch)
fprintf("Damping ratio: %f\n", dampDutch)
fprintf("Time to damp to half the initial amplitude: %f\n", delTDutch)
fprintf("The number of cycles to damp to half the initial amplitude: %f\n",NDutch)

% Spiral
eValSpiral = eVal(:,5);
wdSpiral   = imag(eValSpiral(5));
wnSpiral   = sqrt((real(eValSpiral(5)))^2+(wdSpiral)^2);
dampSpiral = abs(real(eValSpiral(5))/wnSpiral);
delTSpiral = log(2)/abs(real(eValSpiral(5)));
```

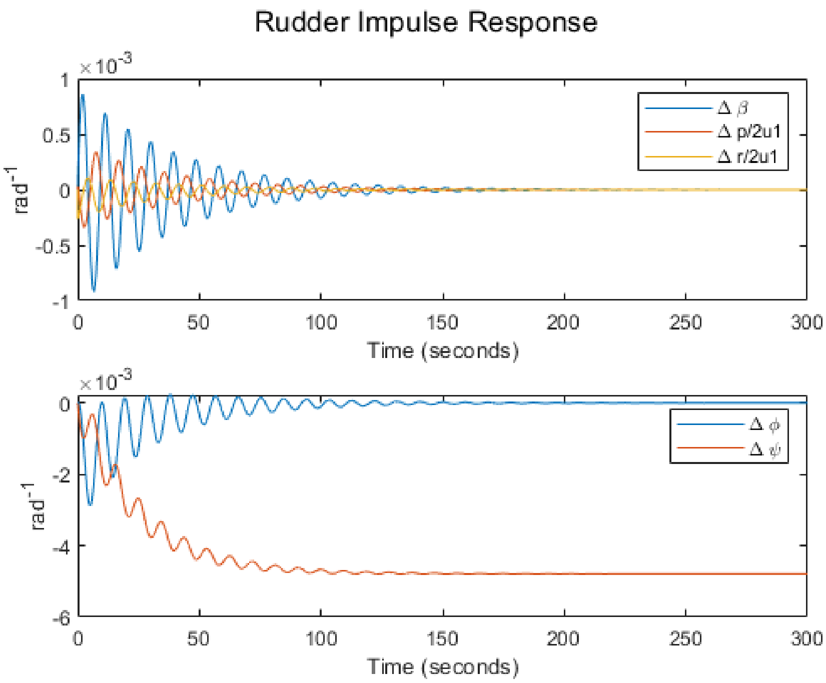
```
NSpiral    = log(2)*wdSpiral/abs(real(eValSpiral(5))*2*pi);

fprintf ("Sprial -------------------------------------------------------")
fprintf ("\nDamped Frequency: %f\n", wdSpiral)
fprintf ("Natural Frequency: %f\n", wnSpiral)
fprintf("Damping ratio: %f\n", dampSpiral)
fprintf("Time to damp to half the initial amplitude: %f\n", delTSpiral)
fprintf("The number of cycles to damp to half the initial amplitude: %f\n",NSpiral)
```

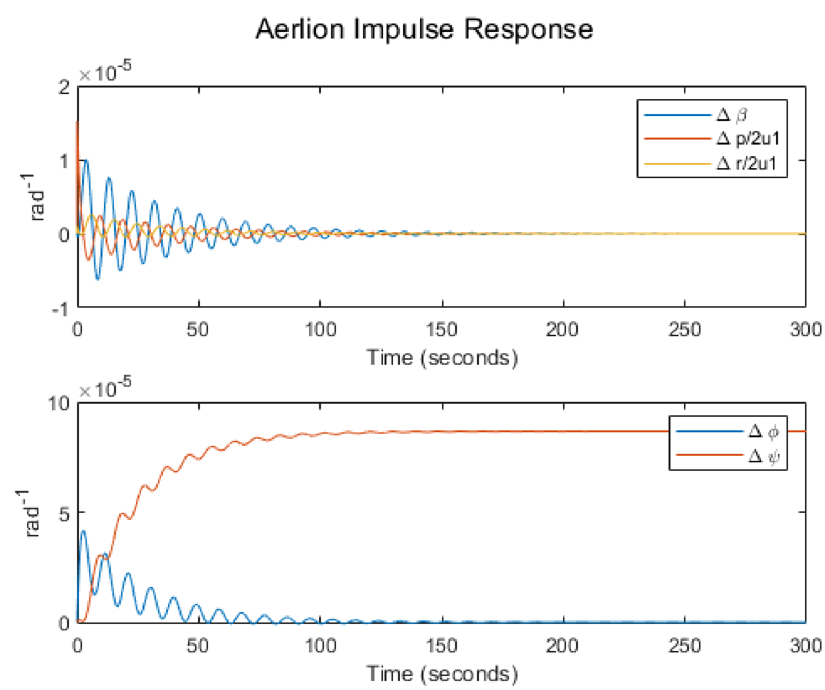
```
Roll---------------------------------------------------------
Damped Frequency: 0.000000
Natural Frequency: 1.053454
Damping ratio: 1.000000
Time to damp to half the initial amplitude: 0.657975
The number of cycles to damp to half the initial amplitude: 0.000000
Dutch Roll---------------------------------------------------
Damped Frequency: 0.672518
Natural Frequency: 0.673027
Damping ratio: 0.038877
Time to damp to half the initial amplitude: 26.490988
The number of cycles to damp to half the initial amplitude: 2.835451
Sprial -------------------------------------------------------
Damped Frequency: 0.000000
Natural Frequency: 0.042649
Damping ratio: 1.000000
Time to damp to half the initial amplitude: 16.252376
The number of cycles to damp to half the initial amplitude: 0.000000
```

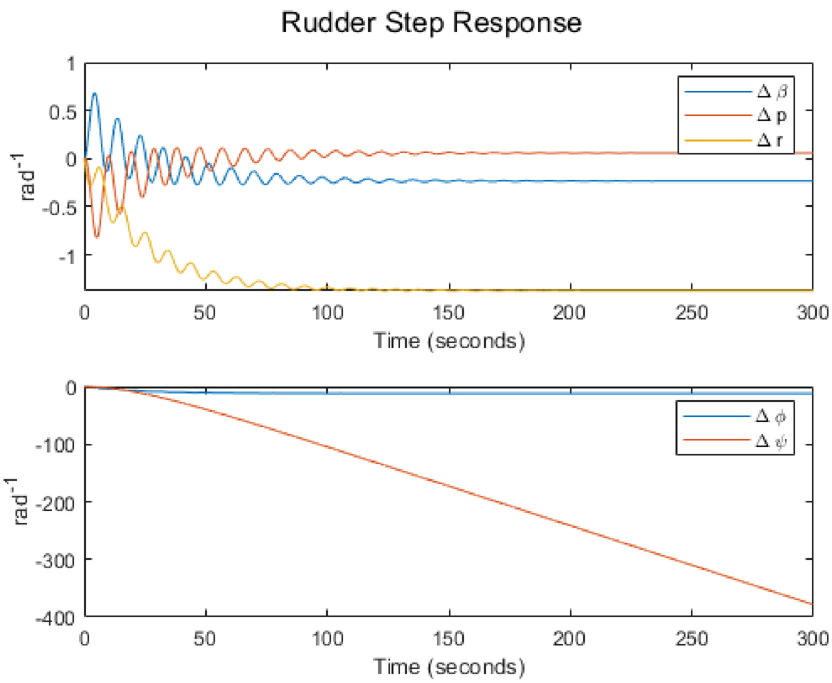
Problem 1 b)

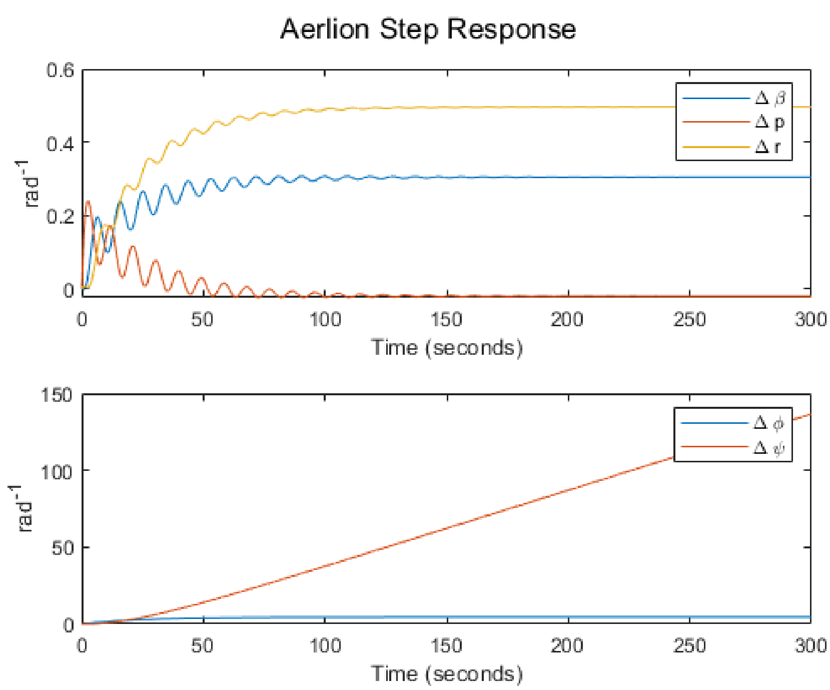
```
sys = ss(A,B,C,D);

t = 0:0.01:300;

figure,
impRud(2,:) = zeros(length(t),1);
impRud(1,:) = zeros(length(t),1);
impRud(2,1) = 20/180*pi; % rad
impRudResponse = lsim(sys,impRud,t);
impRudResponse(:,2) = impRudResponse(:,2)*b/(2*u1);
impRudResponse(:,3) = impRudResponse(:,3)*b/(2*u1);
subplot(2,1,1)
plot(t,impRudResponse(:,1:3))
legend('\Delta \beta', '\Delta p/2u1', '\Delta r/2u1')
ylabel('rad^{-1}')
xlabel('Time (seconds)')
subplot(2,1,2)
plot(t,impRudResponse(:,4:5))
ylabel('rad^{-1}')
xlabel('Time (seconds)')
legend('\Delta \phi', '\Delta \psi')
sgtitle('Rudder Impulse Response')



figure,
impA(2,:) = zeros(length(t),1);
impA(1,:) = zeros(length(t),1);
impA(1,1) = 1/180*pi; % rad
impAResponse = lsim(sys,impA,t);
impAResponse(:,2) = impAResponse(:,2)*b/(2*u1);
impAResponse(:,3) = impAResponse(:,3)*b/(2*u1);
subplot(2,1,1)
plot(t,impAResponse(:,1:3))
legend('\Delta \beta', '\Delta p/2u1', '\Delta r/2u1')
ylabel('rad^{-1}')
```

```
xlabel('Time (seconds)')
subplot(2,1,2)
plot(t,impAResponse(:,4:5))
ylabel('rad^{-1}')
xlabel('Time (seconds)')
legend('\Delta \phi', '\Delta \psi')
sgtitle('Aerlion Impulse Response')


stept = 0:0.01:300;
figure,
stepRud(2,:) = ones(length(stept),1);
stepRud(1,:) = zeros(length(stept),1);
stepRud(2,1:10) = 0;
stepRudResponse = lsim(sys,stepRud,stept);
subplot(2,1,1)
plot(stept,stepRudResponse(:,1:3))
ylabel('rad^{-1}')
xlabel('Time (seconds)')
legend('\Delta \beta', '\Delta p', '\Delta r')
subplot(2,1,2)
plot(stept,stepRudResponse(:,4:5))
ylabel('rad^{-1}')
xlabel('Time (seconds)')
legend('\Delta \phi', '\Delta \psi')
sgtitle('Rudder Step Response')

figure,
stepA(2,:) = zeros(length(stept),1);
stepA(1,:) = ones(length(stept),1);
stepA(1,1:10) = 0;
stepAResponse = lsim(sys,stepA,stept);
subplot(2,1,1)
plot(stept,stepAResponse(:,1:3))
ylabel('rad^{-1}')
xlabel('Time (seconds)')
legend('\Delta \beta', '\Delta p', '\Delta r')
subplot(2,1,2)
plot(stept,stepAResponse(:,4:5))
ylabel('rad^{-1}')
xlabel('Time (seconds)')
legend('\Delta \phi', '\Delta \psi')
sgtitle('Aerlion Step Response')
```


Rudder Impulse Response
×10^-3
rad^-1
Time (seconds)
Δ β
Δ p/2u1
Δ r/2u1
Δ φ
Δ ψ


Aerlion Impulse Response
×10^-5
rad^-1
Time (seconds)
Δ β
Δ p/2u1
Δ r/2u1
Δ φ
Δ ψ





Dutch Roll Approximation

```
A2 = [Yb/u1 Yr/u1-1;
    Nb Nr];
B2 = [Ydr/u1;
   Ndr];
C2 = eye(2);
D2  =zeros(2,1);
[eVec2,eVal2] = eig(A2);
```

Problem 2a)

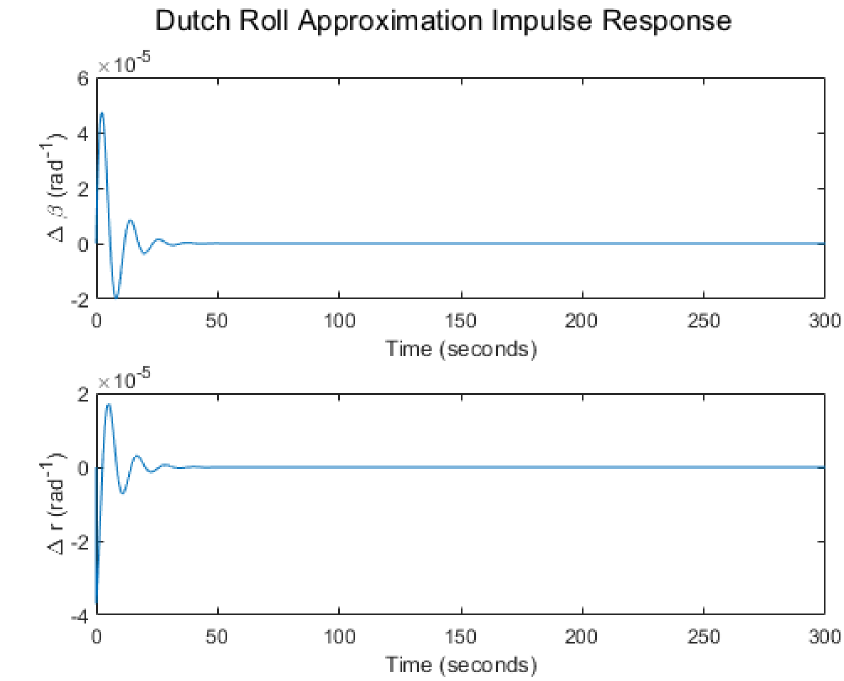
```
wd2 = imag(eVal2(1,1));
% sqrt((real(eVal2(1,1)))^2+(imag(eVal2(1,1)))^2)
% real(eVal2(1,1))/sqrt((real(eVal2(1,1)))^2+(imag(eVal2(1,1)))^2)
```

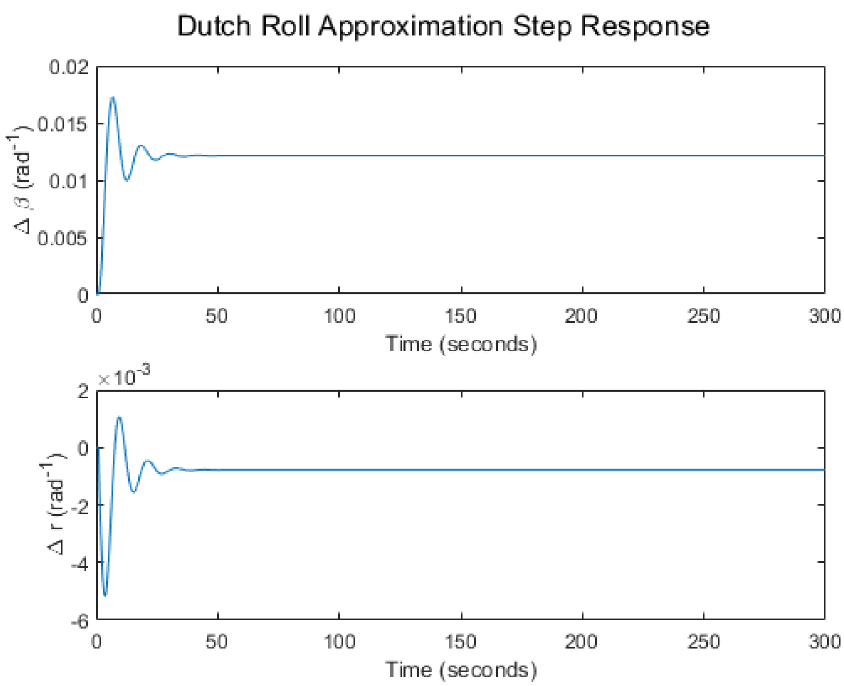
```
wn2 = sqrt((Nr*Yb-Nb*Yr+Nb*u1)/u1);
damp2 = -1/(2*wn2)*(Yb+Nr*u1)/u1;
delT2 = log(2)/abs(real(eVal2(1,1)));
N2    = log(2)*wd2/abs(real(eVal2(1,1))*2*pi);

fprintf ("Dutch Roll Approximation --------------------------------------")
fprintf ("\nDamped Frequency: %f\n", wd2)
fprintf ("Natural Frequency: %f\n", wn2)
fprintf("Damping ratio: %f\n", damp2)
fprintf("Time to damp to half the initial amplitude: %f\n", delT2)
fprintf("The number of cycles to damp to half the initial amplitude: %f\n",N2)
```

```
Dutch Roll Approximation --------------------------------------
Damped Frequency: 0.537090
Natural Frequency: 0.556991
Damping ratio: 0.264924
Time to damp to half the initial amplitude: 4.697385
The number of cycles to damp to half the initial amplitude: 0.401535
```

Problem 2b)

```
sys2 = ss(A2,B2,C2,D2);

t2 = 0:0.01:300;
figure,
imp = zeros(length(t2),1);
imp(1) = 1/180*pi; % rad
impResponse = lsim(sys2,imp,t2);
subplot(2,1,1)
plot(t2,impResponse(:,1))
ylabel('\Delta \beta (rad^{-1})')
xlabel('Time (seconds)')
subplot(2,1,2)
plot(t2,impResponse(:,2))
ylabel('\Delta r (rad^{-1})')
xlabel('Time (seconds)')
sgtitle('Dutch Roll Approximation Impulse Response')


figure,
step = ones(length(t2),1)/180*pi;
step(1:100) = 0; % rad
stepResponse = lsim(sys2,step,t2);
subplot(2,1,1)
plot(t2,stepResponse(:,1))
ylabel('\Delta \beta (rad^{-1})')
xlabel('Time (seconds)')
subplot(2,1,2)
plot(t2,stepResponse(:,2))
ylabel('\Delta r (rad^{-1})')
xlabel('Time (seconds)')
sgtitle('Dutch Roll Approximation Step Response')
```





Problem 3

```
fprintf (['The damped frequency of the dutch roll Mode is %f and that of dutch roll approximate \n' ...
    'is %f. The difference in damped frequency is observed in the plots. The approximate model \n'...
    'shows much less oscillation for both impulse and step response While we can count\n'...
    'a little more than a few oscillation during the first 50 seconds of the approxmimate \n'...
    'model sim for Δβ and Δr, there are definitely more oscillations\n'...
    'observed in first 50 seconds of the simulation for the "Rudder responses" graphs.\n'],wdDutch,wd2)

fprintf (['\nThe natural freqeuncy of of the full model is greater than that of the dutch\n'...
     'roll approximate model by %f Hz.\n'], wnDutch-wn2)

fprintf (['\nThe damping ratio of of the full model is less than that of the dutch\n'...
     'roll approximate model by %f. There fore we see oscillations die much quicker in\n'...
     'the approximation model then those in the full model.\n'], damp2-dampDutch)
```

```
fprintf (['\nThe time to damp of of the full model is greater than that of the dutch\n'...
    'roll approximate model by %f seconds.\n'], delTDutch-delT2)

fprintf (['\nThe number of cycles to damp of of the full model is greater than that of the dutch\n'...
    'roll approximate model by %f.\n'], NDutch-N2)

fprintf (['\nAs I have stated before the oscillations die quicker in the approximation model than the\n'...
    'full model, and this is represented by the larger time and cycle to damp in the full model than the approximate model\n'...
    'This is due to approxmate model''s larger damping ratio than full model''s. \n'])
```

```
The damped frequency of the dutch roll Mode is 0.672518 and that of dutch roll approximate
is 0.537090. The difference in damped frequency is observed in the plots. The approximate model
shows much less oscillation for both impulse and step response While we can count
a little more than a few oscillation during the first 50 seconds of the approxmimate
model sim for Δβ and Δr, there are definitely more oscillations
observed in first 50 seconds of the simulation for the "Rudder responses" graphs.

The natural freqeuncy of of the full model is greater than that of the dutch
roll approximate model by 0.116035 Hz.

The damping ratio of of the full model is less than that of the dutch
roll approximate model by 0.226047. There fore we see oscillations die much quicker in
the approximation model then those in the full model.

The time to damp of of the full model is greater than that of the dutch
roll approximate model by 21.793602 seconds.

The number of cycles to damp of of the full model is greater than that of the dutch
roll approximate model by 2.433916.

As I have stated before the oscillations die quicker in the approximation model than the
full model, and this is represented by the larger time and cycle to damp in the full model than the approximate model
This is due to approxmate model's larger damping ratio than full model's.
```

Published with MATLAB® R2023a