Imperial College London

Advanced Mechanics of Flight

February 10, 2020

Department: Department of Aeronautics
Student: Nathan Davey - 01187454

Question 1

The dynamics of the aircraft are given in state space form with the equation $\dot{x_r} = A_r x_r + B_r \delta_e$. The solution can be determined by linearising the longitudinal and kinematic equations of motion at the nominal decent conditions. The system matrix was calculated as:

$$A_r = \begin{bmatrix} -0.0245 & 0.3021 & 0 & -9.8040 \\ -0.7003 & -1.9123 & 26.8889 & 0.3425 \\ 0 & -0.4513 & -3.7869 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

The stability of the glider could then be determined by calculating the poles λ (eigenvalues) of this state matrix. These are given in the table below:

	Real (λ)	Imaginary (λ)
SPPO ($\lambda_{1,2}$)	-2.8485	$\pm$3.3618
PHUGOID($\lambda_{3,4}$)	-0.0134	$\pm$0.3995

The first pair of complex eigenvalues correspond to the SPPO mode, and the second the phugoid. The presence of an imaginary part indicated oscillatory motion, while the negative real part indicates asymptotic stability, tending to an equilibrium value.

Question 2

The frequency of the Phugoid ω_n is given by the eigenvalues using $\omega_n = \sqrt{Real(\lambda)^2 + Imag(\lambda)^2}$. The period is give as $T = \frac{2\pi}{\omega_n}$.

ω_w (rad/s)	T (s)
0.3995	15.7275

The frequency response to a step input from the elevator is given in figure 1. The key state variables in the phugoid mode are v_x and θ, and these closely match the frequency obtained from the pole analysis.

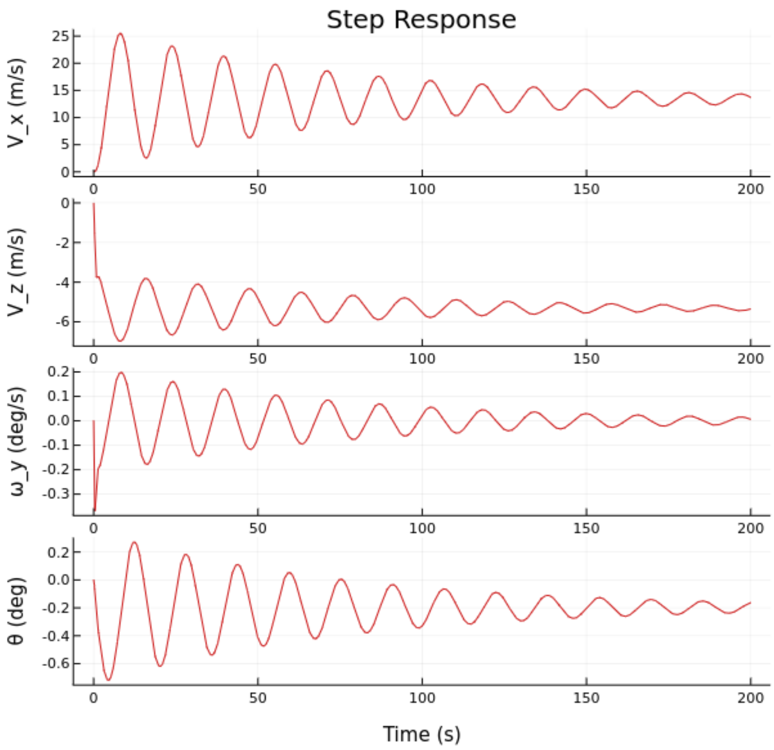

Figure 1: Amplitude responses of an elevator step input.

Question 3

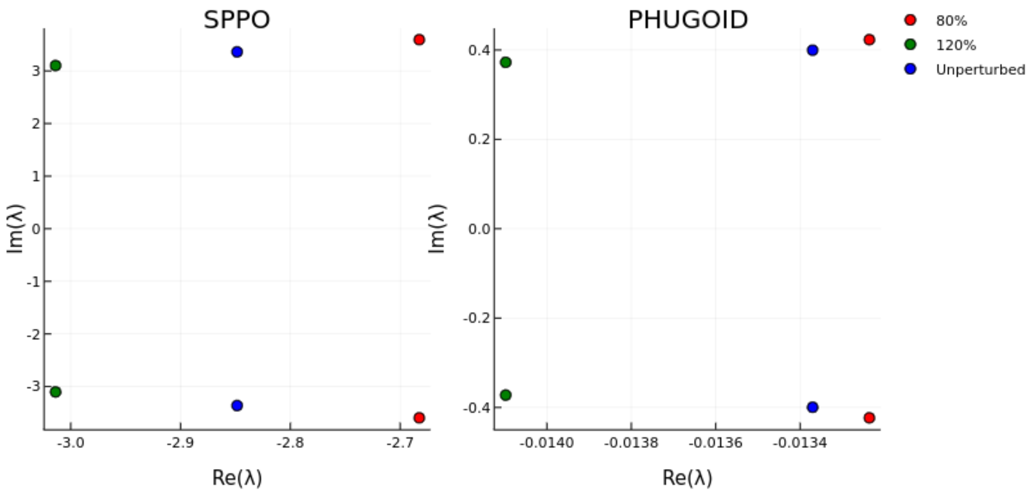

Figure 2: Varying $C^{w}_{L\alpha}$

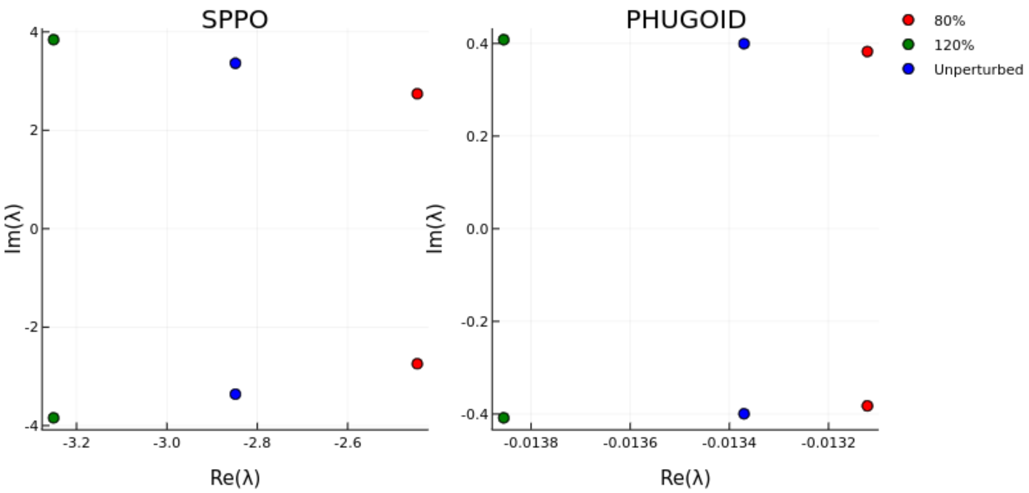

Figure 3: Varying $C^t_{L\alpha}$.

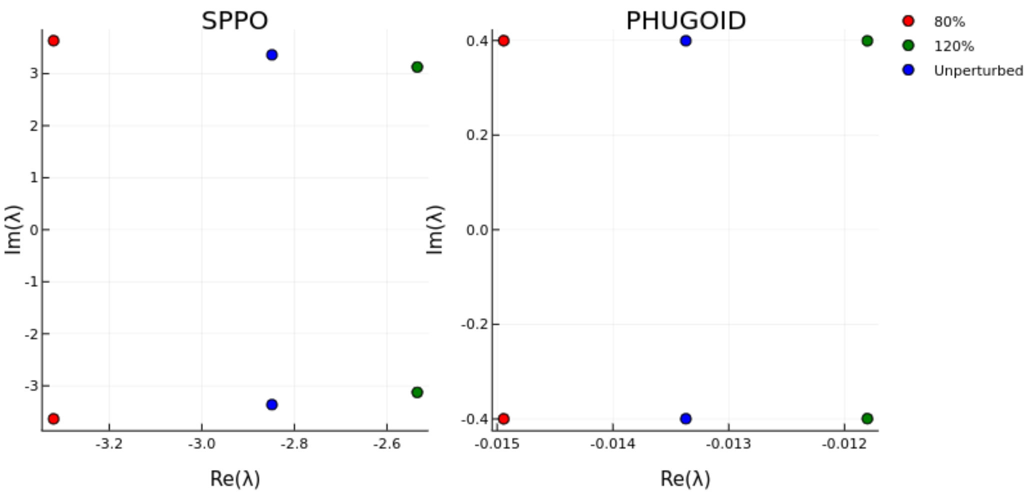

Figure 4: Varying I_{yy}.

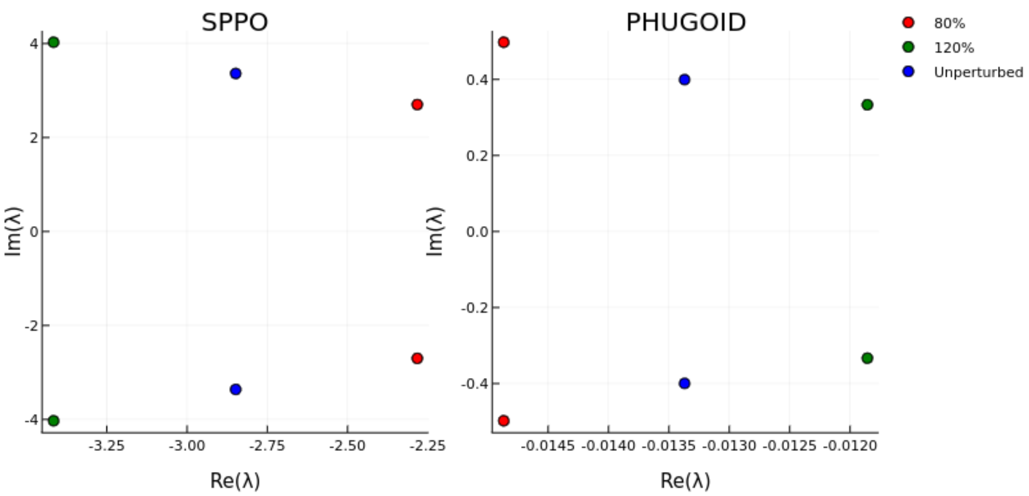

Figure 5: Varying V_∞.

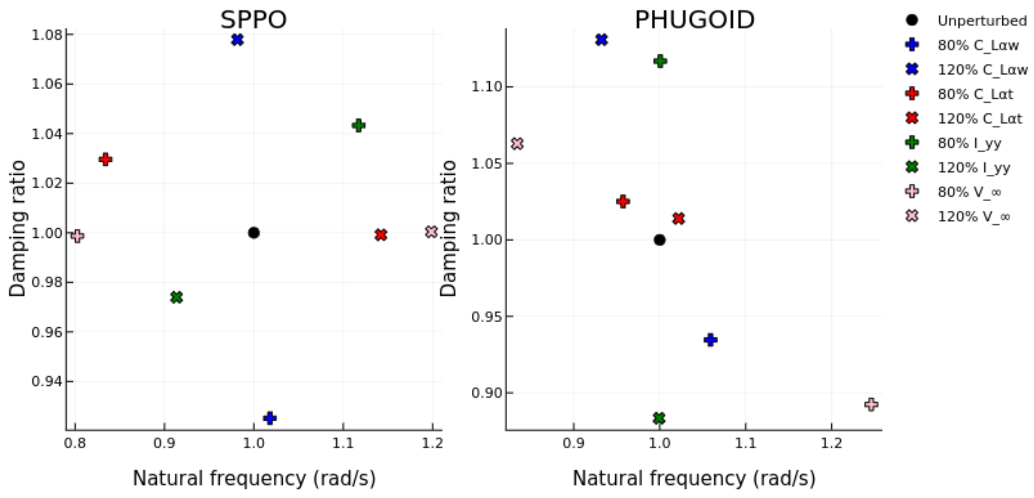

Figure 6: The frequency and damping for each of the cases. The values are normalised by the values of the unchanged system.

The following table illustrates the effect of increasing (opposite effect for decreasing) a given parameter value:

Response to an increase in parameter	$C^w_{L\alpha}$	$C^t_{L\alpha}$	I_{yy}	V_∞
Real part of SPPO	$-$	$-$	$+$	$-$
Imaginary part of SPPO	$-$	$+$	$-$	$+$
Real part of phugoid	$-$	$-$	$+$	$+$
Imaginary part of phugoid	$-$	$+$	$=$	$-$

A decreasing real part indicates the damping of the system is increasing. The effect of varying each parameter is more clearly seen in figure 6. We observe that in SPPO increasing V_∞ has little effect on damping ration but a substantial effect on natural frequency, This is possible due to V_∞ having undergone a Hopf Bifurcation. The opposite is true of I_yy is phugoid, where increasing the parameter has little effect on the natural frequency but a substantial effect on damping, indicating that a further increase could cause a Hopf Bifurcation. Varying $C^t_{L\alpha}$ appears to have little effect in the phugoid mode.

Question 4

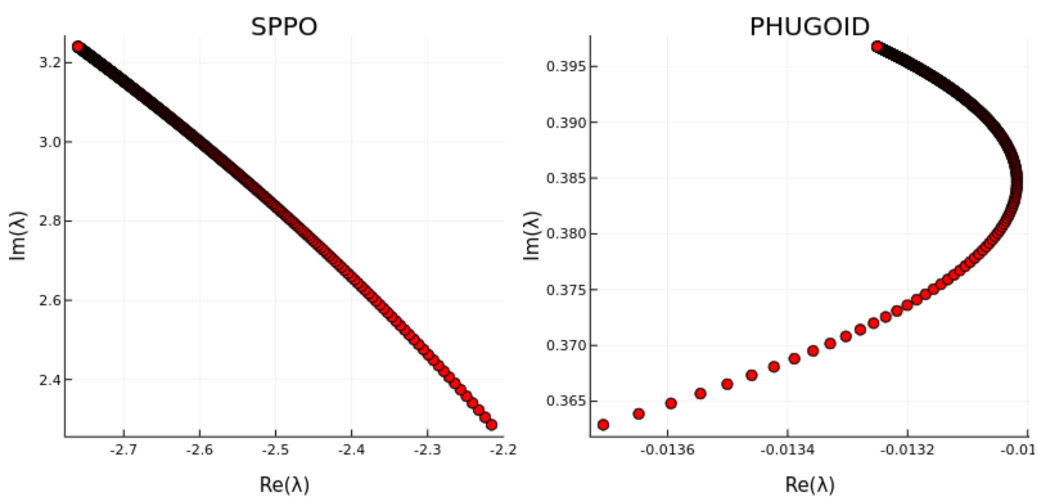

Figure 7: The poles of the system after varying σ between 5 and 50.

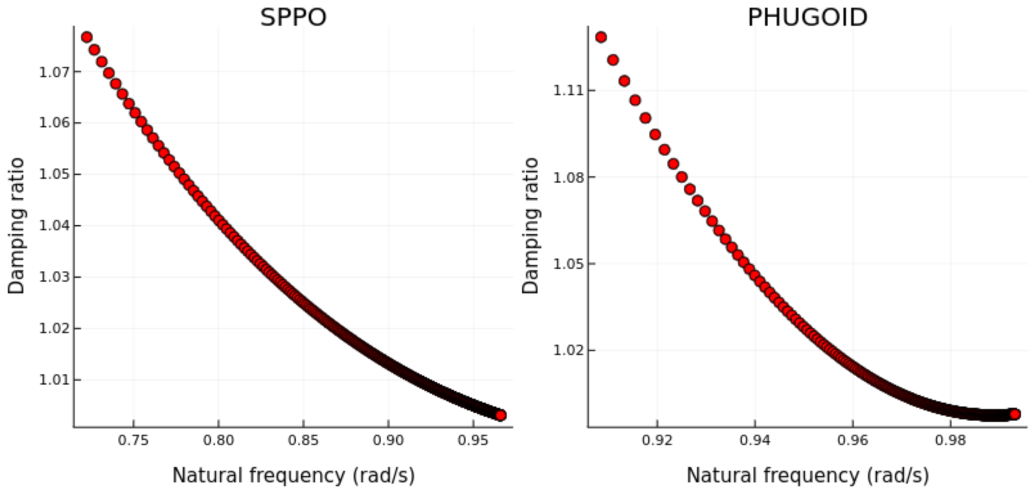

Figure 8: The natural frequency and damping of the system after varying σ between 5 and 50.

We observe in the SPPO mode, increasing stiffness parameter σ increases the stiffness of the fuselage and results in a decrease in the real part, and an increase in the imaginary part. For the phugoid, increasing σ results in an increase in the imaginary part, and an increase in the real part until σ exceeds a value of 10, where it begins to decrease again.

From figure 8 we observe that increasing the stiffness parameter results in decreasing damping ratio and increasing natural frequency for both SPPO and phugoid. This would make sense as we would expect the characteristics to more closely resemble rigid body motion as stiffness is increased. The addition of flexibility results in a more damped oscillation, but too much much flexibility will cause oscillations in the fuselage to match the natural frequency, resulting in a Hopf Bifurcation and the system becoming more excited and unstable.

Question 5 (a)

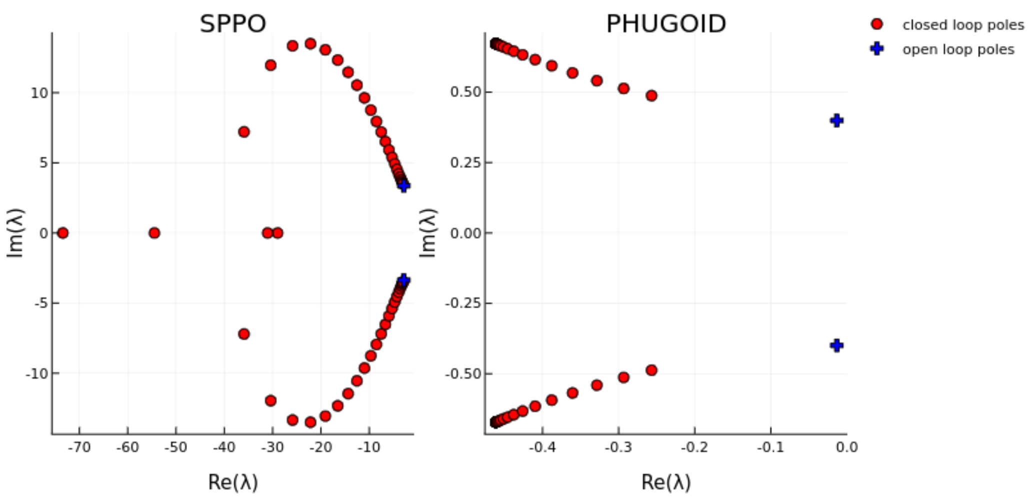

Figure 9: The poles for increasing R. As R increases the poles tend towards the open loop system.

We observe from figure 9 that as the value of R decreases, the poles coalesce into negative real values in the SPPO mode. Increasing R reduces the effectiveness of the elevator in canceling oscillations, and results in a more undamped system as the poles move closer to the real axis.

Question 5 (b)

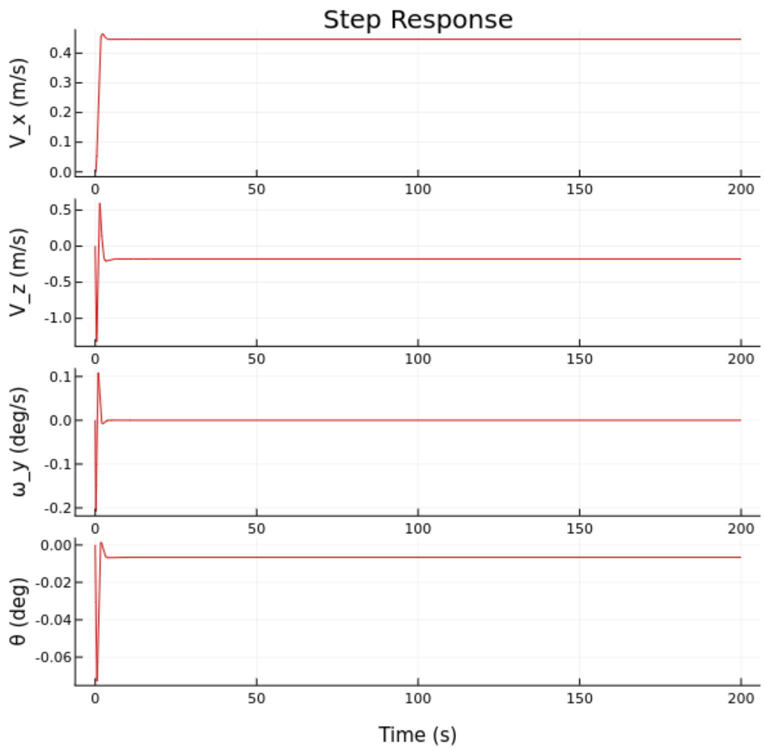

Figure 10: The step response to the chosen Q matrix.

From question 1 we recall that state variables V_x and ω were key variables for the phugoid mode. As such, we should increase the weights of the corresponding Q_{11} and Q_{44} values to increase system damping. The resulting step response for a system where these values are increased is shown in figure 10. It was also observed that varying Q_{22} Q_{33} had a much smaller effect on damping.

```
using Revise
using LinearAlgebra
using ControlSystems
using Plots
using Printf
plotly()
# constant definitions

ρ_∞ = 1.112
θ_0 = -2
V_∞ = 28
m = 318
S = 6.11
l_t = 4.63
I_yy = 432
S_t = 1.14
l_w = 0.3
C_Lαw = 5.55
C_Lαt = 4.3
C_Lδt = 0.45
b = 15
T_0 = 0
e = 0.8
g = 9.81


function getABCD(ρ_∞, θ_0, V_∞, m, S, l_t, I_yy, S_t, l_w,
C_Lαw, C_Lαt, C_Lδt, b, T_0, e, g)
    C_Lα = C_Lαw + S_t * C_Lαt / S
    q_∞ = 0.5 * ρ_∞ * V_∞^2

    # state matrix
    A = [
          [
          2 * ( m * g * sind(θ_0) -T_0) / (m * V_∞)
          g * cosd(θ_0) * (1 - 2 * S * C_Lα / (π * b^2 * e) )
/ V_∞
          0
          -g * cosd(θ_0)
          ]';

          [
          -2 * g * cosd(θ_0) / V_∞
          (m * g * sind(θ_0) - T_0) / (m * V_∞) - ρ_∞ * V_∞ *
S * C_Lα / (2 * m)
          V_∞ - ρ_∞ * V_∞ * S_t * l_t * C_Lαt / (2 * m)
```

```
            -g * sind(θ_0)
            ]';

            [
            0
            ρ_∞ * V_∞ * (S * l_w * C_Lαw - S_t * l_t * C_Lαt) /
(2 * I_yy)
            -ρ_∞ * V_∞ * S_t * l_t^2 * C_Lαt / (2 * I_yy)
            0
            ]';

            [
            0
            0
            1
            0
            ]'
          ]

     # input matrix
      B = [
              0
              -q_∞ * S_t * C_Lδt / m
              -q_∞ * S_t * l_t * C_Lδt / I_yy
              0
           ]

     # sensory matrix
     C = Matrix(I,4,4)

     # direct term
     D = 0
     return A,B,C,D
end

# Question 1
A,B,C,D = getABCD(ρ_∞, θ_0, V_∞, m, S, l_t, I_yy, S_t, l_w,
C_Lαw, C_Lαt, C_Lδt, b, T_0, e, g)

# eigenvalue plots
λ = eigvals(A)
scatter(λ, xlabel="Re(λ)", ylabel="Im(λ)", legend = false)

# Question 2
# state space
sys = ss(A,B,C,D)
stepplot(sys, 200, legend = false)
phugω = imag(λ[4]) # phugiod frequency
@printf("The phugoid frequency is %f and the time period is
```

```
%f", phugω, 2 * π / phugω)

# Question 3
λ_C_Lαw = [
           eigvals(getABCD(ρ_∞, θ_0, V_∞, m, S, l_t, I_yy,
S_t, l_w, C_Lαw*0.8, C_Lαt, C_Lδt, b, T_0, e, g)[1])';
           eigvals(getABCD(ρ_∞, θ_0, V_∞, m, S, l_t, I_yy,
S_t, l_w, C_Lαw*1.2, C_Lαt, C_Lδt, b, T_0, e, g)[1])'
          ]'

λ_C_Lαt = [
           eigvals(getABCD(ρ_∞, θ_0, V_∞, m, S, l_t, I_yy,
S_t, l_w, C_Lαw, C_Lαt*0.8, C_Lδt, b, T_0, e, g)[1])';
           eigvals(getABCD(ρ_∞, θ_0, V_∞, m, S, l_t, I_yy,
S_t, l_w, C_Lαw, C_Lαt*1.2, C_Lδt, b, T_0, e, g)[1])'
          ]'

λ_I_yy = [
           eigvals(getABCD(ρ_∞, θ_0, V_∞, m, S, l_t,
I_yy*0.8, S_t, l_w, C_Lαw, C_Lαt, C_Lδt, b, T_0, e, g)[1])';
           eigvals(getABCD(ρ_∞, θ_0, V_∞, m, S, l_t,
I_yy*1.2, S_t, l_w, C_Lαw, C_Lαt, C_Lδt, b, T_0, e, g)[1])'
          ]'



λ_V_∞ = [
           eigvals(getABCD(ρ_∞, θ_0, V_∞*0.8, m, S, l_t,
I_yy, S_t, l_w, C_Lαw, C_Lαt, C_Lδt, b, T_0, e, g)[1])';
           eigvals(getABCD(ρ_∞, θ_0, V_∞*1.2, m, S, l_t,
I_yy, S_t, l_w, C_Lαw, C_Lαt, C_Lδt, b, T_0, e, g)[1])'
          ]'
q3plot(λ,λ_ref) = scatter([[λ[:,1][1:2];λ[:,2][1:2];
λ_ref[1:2]] [λ[:,1][3:4];λ[:,2][3:4]; λ_ref[3:4]] ],
xlabel="Re(λ)", ylabel="Im(λ)", layout = 2, legend = false,
title=["SPPO" "PHUGOID"])
q3plot(λ_C_Lαw, λ)
q3plot(λ_C_Lαt, λ)
q3plot(λ_I_yy, λ)
q3plot(λ_V_∞, λ)

# Question 4
l_β = (l_w + l_t) / 2 # position of spring
σ = 5:0.1:50
k_β = 0.5 * ρ_∞ * V_∞^2 * S_t * l_t * σ

A_s = -0.5 * ρ_∞ * V_∞ ^2 * S_t * l_β * C_Lαt
A_rs = ρ_∞ * V_∞^2 * S_t * C_Lαt / (2 * m) * [0,1,l_t * m /
I_yy,0]
A_sr = 0.5 * ρ_∞ * V_∞ * S_t * l_β * C_Lαt * [0 1 l_t 0]
```

```
A_k = [A + A_rs * A_sr / (k_β[i] - A_s) for i =
1:length(k_β)] # state matrix with k spring
λ_k = map((x) -> eigvals(x), A_k)

# get modes
sppo_k = begin
    l = length(λ_k)
    [[λ_k[i][1] for i = 1:l]; [λ_k[i][2] for i = 1:l]]
end

phug_k = begin
    l = length(λ_k)
    [[λ_k[i][3] for i = 1:l]; [λ_k[i][4] for i = 1:l]]
end

# plot poles
scatter([sppo_k phug_k], xlabel="Re(λ)", ylabel="Im(λ)",
layout = 2, legend = false)
```