```
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)
```