AS6320 ASSIGNMENT 3

Shwetha V ME23B073

April 2025

1 Introduction

The occurrence of sustained large amplitude oscillations in engines is very detrimental to their performance, and this phenomenon called as 'Thermo-acoustic instability'. This is because the instability arises due to a positive feedback and coupling between the acoustic pressure(sound), heat release(combustion) and hydrodynamics(fuel flow). Hence it is very important to study and understand the physics behind this phenomena, so that safe engines can be designed and damage can be prevented.

In this report, thermoacoustic interaction in a Rijke tube is studied using a model, which matches experimental results.

2 The Model used

The model used is a 1-dimensional linear model.

The temporal evolution of the acoustic perturbations is studied using Galerkin technique. The fluctuating heat release from the heating element is treated as a compact source, and is modeled by a low order time lag model, given by Heckl based on a modified form of King's law.

The Rijke tube model thus obtained is a physical model which is based on experimental observations.

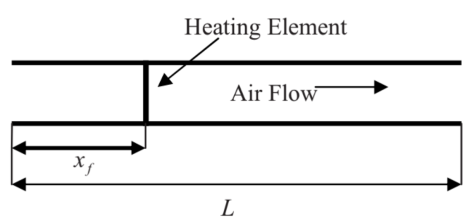

Figure 1: Schematic of the horizontal Rijke tube setup

2.1 Governing Equations

The governing equations for the 1-D acoustic field, neglecting the effects of mean flow and mean temperature are: Conservation of acoustic momentum

$$\bar{\rho}\frac{\partial \tilde{u}'}{\partial \tilde{t}} + \frac{\partial \tilde{p}'}{\partial \tilde{x}} = 0 \tag{1}$$

Conservation of acoustic energy

$$\frac{\partial \tilde{p}'}{\partial \tilde{t}} + \gamma \bar{p}\frac{\partial \tilde{u}'}{\partial \tilde{x}} = (\gamma - 1)\,\dot{\tilde{Q}}' \tag{2}$$

A modified form of King's law is used to model the heat release rate.

$$\dot{\tilde{Q}}' = \frac{2L_w(T_w - \bar{T})}{S\sqrt{3}}\sqrt{\pi\lambda C_v \bar{\rho}\frac{d_w}{2}}\left[\sqrt{\left|\frac{u_0}{3} + u'_f(t-\tau)\right|} - \sqrt{\frac{u_0}{3}}\right]\delta(\tilde{x} - \tilde{x_f}) \tag{3}$$

where,
L_w is the equivalent length of the wire,
λ is the heat conductivity of air,
C_v is the specific heat of air at constant volume,
τ is the time lag,
$\bar{\rho}$ is the mean density of air,
d_w is the diameter of the wire,
$T_w - \bar{T}$ is the temperature difference and
S is the cross-sectional area of the duct.
The above equations can be non-dimensionalized as follows:

$$\tilde{x} = L_a x, \quad \tilde{t} = \frac{L_a}{c_o}t, \quad \tilde{u}', = u_0 u', \quad \tilde{p} = \bar{p}p', \quad M = \frac{u_0}{c_0}, \tag{4}$$

where,
c_0 is the speed of sound,
L_a is the duct length,
$\bar{p}$ is the pressure of the undisturbed medium and
u_0 is the mean flow velocity.

The acoustic equations in non-dimensional form can be written as follows:

$$\gamma M \frac{\partial u'}{\partial t} + \frac{\partial p'}{\partial x} = 0, \tag{5}$$

$$\frac{\partial p'}{\partial t} + \gamma M \frac{\partial u'}{\partial x} = K \left[\sqrt{\left| \frac{1}{3} + u'(t-\tau) \right|} - \sqrt{\frac{1}{3}} \right] \delta(x - x_f) \tag{6}$$

where,

$$K = (\gamma - 1) \frac{2L_w(T_w - \bar{T})}{Sc_0 \bar{p} \sqrt{3}} \sqrt{\pi \lambda C_v \bar{\rho} \frac{d_w}{2} u_0} \tag{7}$$

The above set of partial differential equations can be reduced to ODEs using the Galerkin technique. The velocity and pressure field can be written in terms of the duct's natural modes as follows:

$$u' = \sum_{j=1}^{\infty} \eta_j cos(j\pi x) \quad and \quad p' = -\sum_{j=1}^{\infty} \frac{\gamma M}{j\pi} \dot{\eta}_j sin(j\pi x). \tag{8}$$

Any function in a domain, can be expressed in terms of a set of basis functions of that domain. The Galerkin technique makes use of this principle, and the basis functions are chosen such that they satisfy the boundary conditions. By substituting the velocity and pressure field expression into the non-dimensionalized equations obtained earlier and projecting along the basis functions($\eta, \dot{\eta}$), The following evolution equations are obtained:

$$\frac{d\eta_j}{dt} = \dot{\eta}_j, \tag{9}$$

$$\frac{d\dot{\eta}_j}{dt} + 2\zeta_j \omega_j \dot{\eta}_j + k_j^2 \eta_j = -2j\pi K \left[\sqrt{\left| \frac{1}{3} + u'_f(t-\tau) \right|} - \sqrt{\frac{1}{3}} \right] sin(j\pi x_f). \tag{10}$$

Here, $2\omega_j \zeta_j$ is the expression for frequency dependent damping where ζ_j is given by

$$\zeta_j = \frac{1}{2\pi} \left(c_1 \frac{w_j}{w_1} + c_2 \sqrt{\frac{w_1}{w_j}} \right), \tag{11}$$

where c_1 and c_2 are the damping coefficients which can be varied to control the amount of damping in the system.

The obtained non-linear coupled first order ODEs are solved by numerical integration, and evolution of the acoustic velocity for different modes and the corresponding energy diagram is studied. The effect of varying the non-dimensionalized heater power (K) on the evolution of the system is analyzed.

3 Numerical method: Explanation of the Fourth Order Runge-Kutta integration scheme

The Runge-Kutta method is used to find approximate values of y for given values of x, of an ordinary differential equation of the form $\frac{dy}{dx}$ when the initial value of y i.e., y(0) is given. $\frac{dy(t)}{dx} = y'(t) = f(y(t), t), \quad \text{with} \quad y(t_0) = y_0$

The basic forward Euler method uses the information of the slope or derivative of y at a given time step to extrapolate the solution to the next time-step. $y(t_0 + h) \approx y(t_0) + f(y(t_0), t_0)h$

The RK4 method uses information of the slope at more than one point to extrapolate the solution to a future time step. Only first-order ordinary differential equations can be solved by using this method.

The below algorithm is used to compute the next value.

Starting from some known initial condition $y(t_0) = y_0$, We will use the following slope approximations to estimate the slope at some time t_0 (assuming we only have an approximation to $y(t_0)$ which we call $y*(t_0)$).

$k_1 = f(y*(t_0), t_0)$
$k_2 = f(y*(t_0) + k_1\frac{h}{2}, t_0 + \frac{h}{2}$ $k_3 = f(y*(t_0) + k_2\frac{h}{2}, t_0 + \frac{h}{2}$
$k_4 = f(y*(t_0) + k_3h, t_0 + h)$
Each of these slope estimates can be described verbally.

- k_1 is the slope at the beginning of the time step.
- If we use the slope k_1 to step halfway through the time step, then k_2 is an estimate of the slope at the midpoint
- If we use the slope k_2 to step halfway through the time step then k_3 is another estimate of the slope at the midpoint.
- Finally, we use the slope k_3 to step all the way across the time step to $t_0 + h$, and k_4 is an estimate of the slope at the endpoint.

We then use a weighted sum of these slopes to get our final estimate of $y*(t_0+h)$
$y*(t_0 + h) = y*(t_0) + \frac{k_1+2k_2+2k_3+k_4}{6}h$
$= y*(t_0) + (\frac{1}{6}k_1 + \frac{1}{3}k_2 + \frac{1}{3}k_3 + \frac{1}{6}k_4)h$
$= y*(t_0) + mh$
where m is a weighted average slope approximation

4 Results and Discussion

The parameters used in the model are: $c_1 = 0.1, \quad c_2 = 0.06$, time lag τ and Heater power K.
c_1 and c_2 are damping coefficients which can be varied to control the amount of damping in the system.

Here we can see the decay of oscillations for a value of the heater power $K_1 = 0.4$ and growth for a different initial condition $K_2 = 0.8$.

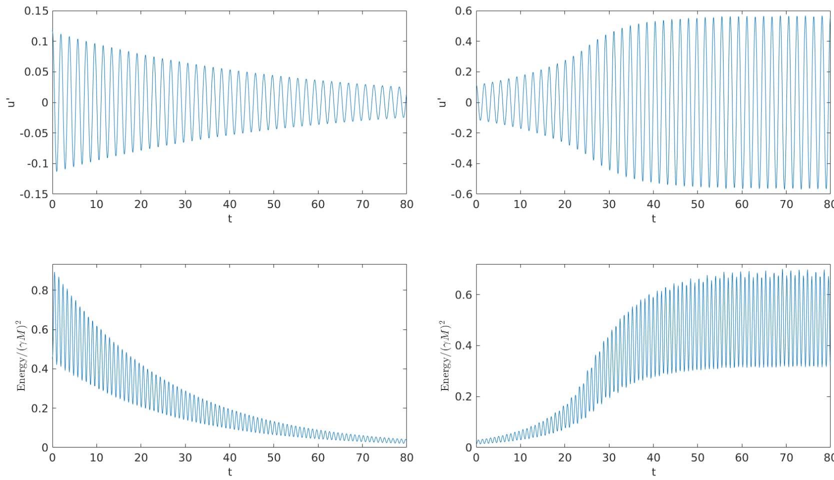

Figure 2: Evolution of non-dimensional acoustic velocity and acoustic energy when the initial condition is $\eta_1(0) = 0.2, \tau = 0.2, K_1 = 0.4$(decay) and $K_2 = 0.8$ (growth)

Further, the figure shows that the amplitude of the oscillations saturate in the case of growth. The evolution of the nondimensional acoustic energy $[\frac{1}{2}p'^2 + \frac{1}{2}(\gamma M u')^2]$ after locally weighted regression smoothing calculated from the equations is plotted. The figure shows the decay of the acoustic energy during decay of velocity oscillations and growth plus saturation of the acoustic energy in the case of growth.

Figure 3 shows the evolution of the acoustic velocity at the heater location $x_f = 0.3$. The oscillations saturate after nonlinear growth. Further the projection of the acoustic velocity on the first three Galerkin expansion functions is plotted.

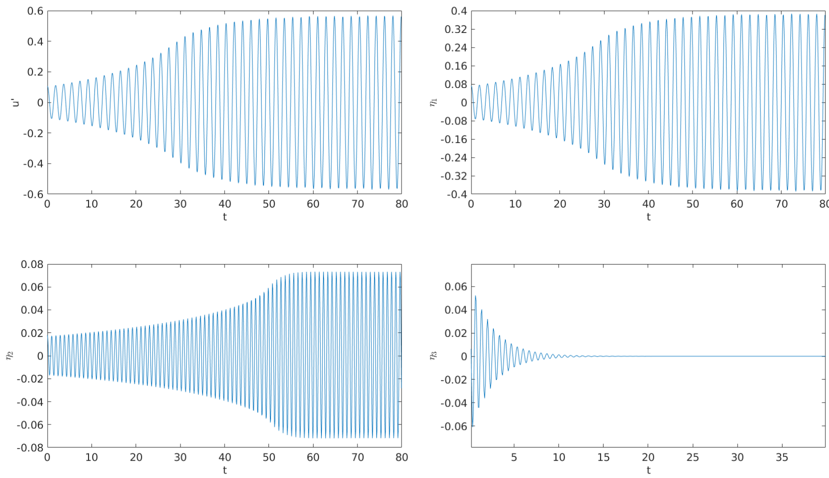

Figure 3: Non-dimensional evolution of acoustic velocity when the initial condition is $\eta_1 = 0.2, \tau = 0.2$ and evolution of the acoustic velocity projected onto the various Galerkin modes when K = 0.8 (growth)

An increase in K represents an increase in the driving force given to the system. Increased driving strives to destabilize the system. Therefore, for small values of K the equilibrium is stable and all perturbations decay asymptotically to zero. Increasing K decreases the margin of stability of the system and at a critical value of K, a pair of complex eigenvalues of the system cross over to the right half plane(Hopf bifurcation) and the system becomes linearly unstable resulting in an oscillating flow pattern in the tube. Limit cycle oscillations are obtained. The bifurcation is subcritical and the resulting small-amplitude limit cycles close to the Hopf point(point where the abrupt jump in U rms takes place) are unstable. This unstable branch of limit cycles undergo a fold or turning point bifurcation and gains stability.

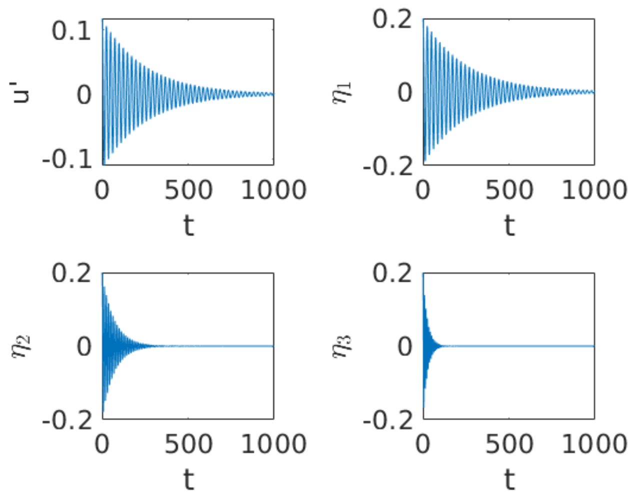

Figure 4: Evolution of the acoustic velocity and its projection on various Galerkin modes when K = 0.2 (decay)

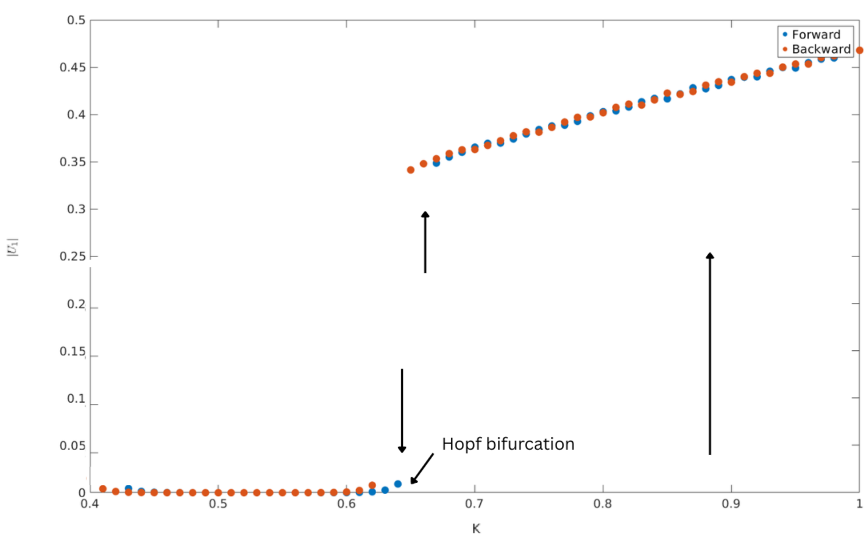

Figure 5: Bifurcation plot for variation of non-dimensional heater power K. The plot shows the occurrence of a Subcritical bifurcation.

From the 3D bifurcation plot we can see the variation in the value of the critical heater power with varying time lag values.

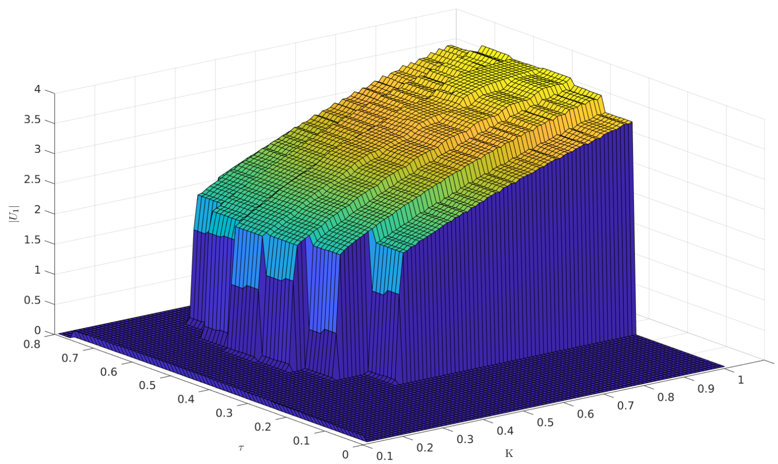

Figure 6: 3-D Bifurcation plot for variation of non-dimensional heater power K for varying values of time lag τ

5 Conclusion

Hence the nature of thermoacoustic interaction is studied using a simple model for a horizontal Rijke tube along with Bifurcation analysis.

6 References

1. Koushik Balasubramanian, R. I. Sujith; Thermoacoustic instability in a Rijke tube: Non-normality and nonlinearity. Physics of Fluids 1 April 2008; 20 (4): 044103. https://doi.org/10.1063/1.2895634.

2. Subramanian, Priya & Mariappan, Sathesh & Sujith, Raman & Wahi, Pankaj. (2010). Bifurcation Analysis of Thermoacoustic Instability in a Horizontal Rijke Tube. International Journal of Spray and Combustion dynamics. 2. 325-356. 10.1260/1756-8277.2.4.325.

7 Code

```
function xout = rk4singlestep(fun,dt,tk,xk)
f1 = fun(tk,xk);
f2 = fun(tk+dt/2, xk+(dt/2)*f1);
f3 = fun(tk+dt/2, xk+(dt/2)*f2);
f4 = fun(tk+dt, xk+dt*f3);
```

```
xout = xk+(dt/6)*(f1+2*f2+2*f3+f4);
```

Listing 1: Function defining the RK4

```
function deta1=galerkin1(t,eta,z,w,k)
deta1=[eta(2), -2*z*w*eta(2)-k^2*eta(1)];

function deta_t=galerkin_t(t,eta,z,w,k,j,K,i,tau,dt,u,xf)
deta_t=[eta(2), -2*z*w*eta(2)-k^2*eta(1)-2*j*pi*K*(sqrt(abs(1/3+u(
    int32(i-tau/dt))))-sqrt(1/3))*sin(j*pi*xf)];
```

Listing 2: Function defining the system equations

```
clc
clear all
close all
%%
T=80;
N=1000;
dt=T/N;
t=0:dt:T-dt;
j=1;

tau=0.2;
c1=0.1;
c2=0.06;
xf=0.3;

w1=1*pi;
wj=j*pi;
kj=j*pi;
K=0.4;
gamma=1.4;
M=0.005;

z=1/(2*pi)*(c1*wj/w1+c2*sqrt(w1/wj));


eta0=[1 0];
eta=eta0;

u=eta(1)*cos(pi*xf);
p=-gamma*M/(j*pi)*eta(2)*sin(j*pi*xf)
for i=2:length(t)

        if t(i)<tau
            fun=@(t,eta) galerkin1(t,eta,z,wj,kj);
        else
            disp(i)
            fun=@(t,eta) galerkin_t(t,eta,z,wj,kj,j,K,i,tau,dt,u,xf
    );
        end
        eta(i,:)=rk4singlestep(fun,dt,t(i),eta(i-1,:));
        u=[u eta(i,1)*cos(pi*xf)];
        p=[p -gamma*M/(j*pi)*eta(i,2)*sin(j*pi*xf)];
end

% figure;
```

```
% plot(u)
% figure;
% plot(p)
% figure;
% plot(eta)

E=1/2*p.^2 + 1/2*(gamma*M*u).^2;

figure;
tiledlayout(2,2,TileIndexing="columnmajor");
nexttile
plot(u)
xlabel('time');
ylabel("u'");
nexttile
plot(smooth(t,E,'loess'))
%% Bifurcation plot
b_fwd=rms(u);
Kay=0.4:0.01:1;
Eta=zeros(length(Kay),1000,2);
index=1;
Eta(index,:,:)=eta;
eta=eta(end,:);
for K=0.41:0.01:1
    u=eta(1)*cos(pi*xf);
    p=-gamma*M/(j*pi)*eta(2)*sin(j*pi*xf);
    for i=2:length(t)
        if t(i)<tau
            fun=@(t,eta) galerkin1(t,eta,z,wj,kj);
        else
            disp(i)
            fun=@(t,eta) galerkin_t(t,eta,z,wj,kj,j,K,i,tau,dt,u,xf);
        end
        eta(i,:)=rk4singlestep(fun,dt,t(i),eta(i-1,:));
        u=[u eta(i,1)*cos(pi*xf)];
        p=[p -gamma*M/(j*pi)*eta(i,2)*sin(j*pi*xf)];
    end
    b_fwd=[b_fwd rms(u)];
    index=index+1;
    Eta(index,:,:)=eta;
    eta=eta(end,:);
end
b_bwd=rms(u);

eta=Eta(end,end,:);
index = 0;
for K=0.99:-0.01:0.40
    u=eta(1)*cos(pi*xf);
    p=-gamma*M/(j*pi)*eta(end,2)*sin(j*pi*xf);
    for i=2:length(t)
        if t(i)<tau
            fun=@(t,eta) galerkin1(t,eta,z,wj,kj);
        else
            disp(i)
            fun=@(t,eta) galerkin_t(t,eta,z,wj,kj,j,K,i,tau,dt,u,xf);
```

```
        end
        eta(i,:)=rk4singlestep(fun,dt,t(i),eta(i-1,:));
        u=[u eta(i,1)*cos(pi*xf)];
        p=[p -gamma*M/(j*pi)*eta(i,2)*sin(j*pi*xf)];
    end
    b_bwd=[b_bwd rms(u)];
    index = index+1;
    eta=Eta(end-index,end,:);
end

figure;
s1=scatter(0.4:0.01:1,b_fwd,14,'filled')
hold on
s2=scatter(0.4:0.01:1,fliplr(b_bwd),14,'filled')
hold off
legend('Forward','Backward')
```

Listing 3: Main code

```
clc
clear all
close all
%%
T=80;
N=1000;
dt=T/N;
t=0:dt:T-dt;
j=1;
c1=0.1;
c2=0.06;
xf=0.3;

w1=1*pi;
wj=j*pi;
kj=j*pi;
K=0.4;
gamma=1.4;
M=0.005;

z=1/(2*pi)*(c1*wj/w1+c2*sqrt(w1/wj));


eta0=[0.2 0];
%% Bifurcation plot

i_tau=1;
Tau=0.01:0.01:0.8;
Heat_P=0.1:0.01:1;
b_fwd=zeros(length(Tau),length(Heat_P));
for tau=0.01:0.01:0.8
    try
        disp(tau)
        eta=eta0;
        i_K=1;
        for K=0.1:0.01:1
            u=eta(1)*cos(pi*xf);
            p=-gamma*M/(j*pi)*eta(2)*sin(j*pi*xf);
            for i=2:length(t)
```

```
                if t(i)<tau
                    fun=@(t,eta) galerkin1(t,eta,z,wj,kj);
                else
                    % disp(i)
                    fun=@(t,eta) galerkin_t(t,eta,z,wj,kj,j,K,i,tau
    ,dt,u,xf);
                end
                eta(i,:)=rk4singlestep(fun,dt,t(i),eta(i-1,:));
                u=[u eta(i,1)*cos(pi*xf)];
                p=[p -gamma*M/(j*pi)*eta(i,2)*sin(j*pi*xf)];
            end
            b_fwd(i_tau,i_K)=rms(u);
            i_K=i_K+1;
            eta=eta(end,:);
        end
        i_tau=i_tau+1;
    catch
        continue
    end
end
```
Listing 4: Code for 3D bifurcation plot