

A Bayesian Approach for Predicting Thermoacoustic Oscillations in a Rijke Tube



Francesco Garita

Department of Engineering
University of Cambridge

First-Year Report

Supervisor: Prof. Matthew P. Juniper

*To my grandparents,
wherever they are.*

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this first-year report are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This first-year report is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This first-year report contains fewer than 15,000 words including appendices, bibliography, footnotes, tables and equations.

Francesco Garita

*Cambridge,
November 2018*

Acknowledgements

First of all I would like to thank my supervisor, Prof. Matthew Juniper, for his everyday support. His encouraging and stimulating words are extremely helpful to me, especially when technical problems arise, which often occurs in research. I consider myself extremely lucky of having come across a supervisor who values human relationships before any working relationships, and to him goes my eternal gratitude. I also need to thank my friend and colleague Hans Yu for being available at any time for technical discussions. His determination, together with his passion for science and technology, motivates myself to do better and better. In addition to him, I am extremely thankful to my other colleagues, Jack Brewster, Max Croci, Stefano Falco, Filip Gökstorp, Petr Kungurtsev, and Ubaid Qadri, as well as my friend Pau Farres, for the everyday discussions that span over a wide range of topics. Moreover, I would like to extend my acknowledgements to every single member of the Hopkison Lab, particularly to the lab technicians. Last but not least, I need to thank my parents for their support, thanks to which I manage to carry on and face any difficulty.

Abstract

Predicting and eliminating thermoacoustic oscillations is a significant challenge in gas turbine design. Current techniques often fail at predicting the onset of thermoacoustic instabilities because of their extremely high sensitivity to unknown model parameters. In the present study, we combine a thermoacoustic experiment with a low order thermoacoustic model and use data assimilation to infer the parameters of the model, rendering it predictive over some parameter range. The experiment is a vertical Rijke tube containing an electric heater (up to 300 Watts). The heater drives a base flow via natural convection, and thermoacoustic oscillations via velocity-driven heat release fluctuations. The decay rates and frequencies of these oscillations are measured every few seconds. There are two models: one for the base flow and one for the acoustics, both of which are unsteady. The parameters of the base flow model (Nusselt numbers and pressure loss coefficient) are estimated from many thousand measurements using Bayesian regression. For this an Ensemble Kalman Filter is used, which takes into account both model and experimental uncertainties. The parameters of the acoustic model (interaction index and time delay) are inferred by simple regression. Boundary conditions are computed using the multi-microphone method, which relies on both experimental observations and base flow model predictions. The process reveals an extremely high sensitivity of the growth rate predictions to the boundary conditions. This indicates that the multi-microphone method should also be treated in a probabilistic fashion, as are the base flow model parameters.

Keywords: Thermoacoustic Oscillations, Data Assimilation, Ensemble Kalman Filter.

Table of Contents

List of Figures	xiii
List of Tables	xv
Nomenclature	xvii
1 Introduction	1
2 Background theory	5
2.1 Base flow model	5
2.1.1 Model	5
2.1.2 Non-dimensionalisation	7
2.1.3 Summary of the non-dimensional model	8
2.1.4 Numerical implementation	8
2.2 Ensemble Kalman Filter	11
2.3 Experimental setup and data acquisition	17
2.3.1 Apparatus	17
2.3.2 Data acquisition	19
2.4 Data processing	20
2.4.1 Experimental decay rate and frequency	20
2.4.2 Multi-microphone method	21
2.5 Thermoacoustic model	23
2.6 Problem setting	28
2.7 Summary	31
3 Results	33
3.1 Data assimilation	33
3.2 Experimental eigenvalues	37
3.3 Evaluation of the parameters of the acoustic model	38
4 Conclusions and future work	47
Appendices	49
References	59

List of Figures

2.1	Shape of the source term normalized such that it integrates to 1. Parameters of the Gaussian distribution are $\mu = 0.25$, $\sigma^2 = 0.001$.	10
2.2	Sketch of the experimental setup. An electric heater is placed at one quarter of the tube length from the bottom. Six microphones are distributed along the tube, and a loudspeaker is placed at the bottom. Thermocouples, not shown here, are installed inside and on the external surface of the tube at different locations.	18
2.3	(a) 50 ms long sinusoidal signal provided to the system; (b) filtered signal: exponential decay with best-fit line in red	21
2.4	Tube of length L open at both ends with unknown reflection coefficient R at $x = 0$, excited by an acoustic pulse at $x = -L$. . .	21
2.5	Network model of a tube of length L with a heat source located at $x = b$. The arrows represent waves travelling at the local speed of propagation. These waves reflect off the upstream and downstream ends, characterised by reflection coefficients R_u and R_d , respectively.	24
3.1	Solid (top) and gas (bottom) temperature profiles along the tube axis at $t = 14.2$ h (corresponding to $q = 120$ W). Exp. 1 (left): $T_a = 300.5$ K and heater from the top; Exp. 2 (right): $T_a = 297.5$ K and heater from the bottom. Circles are experiments, solid lines are model predictions	34
3.2	Error bar of size $\pm 2\sigma$ of the base flow model parameters. Exp. 1 (left): $T_a = 300.5$ K and heater from the top; Exp. 2 (right): $T_a = 297.5$ K and heater from the bottom	35
3.3	Inlet gas velocity and average speed of propagation upstream and downstream of the heater. Exp. 1 (left): $T_a = 300.5$ K and heater from the top; Exp. 2 (right): $T_a = 297.5$ K and heater from the bottom	36
3.4	Experimental growth rate (top) and frequency (bottom) of the oscillations. Exp. 1 (left): $T_a = 300.5$ K and heater from the top; Exp. 2 (right): $T_a = 297.5$ K and heater from the bottom	37
3.5	Experimental observations and model predictions of growth rate (top) and frequency (bottom) of the oscillations, using Mic. 1, 2, 3, 4, 5. Exp. 1 (left): $T_a = 300.5$ K and heater from the top; Exp. 2 (right): $T_a = 297.5$ K and heater from the bottom. Best-fit parameters for Exp. 1: $k_n = 21.01$ and $k_\tau = 10.84$. Best-fit parameters for Exp. 2: $k_n = 169.94$ and $k_\tau = 8.95$	39

3.6	Experimental observations and model predictions of growth rate (top) and frequency (bottom) of the oscillations, using Mic. 1, 2, 3, 4, 5, 6. Exp. 1 (left): $T_a = 300.5$ K and heater from the top; Exp. 2 (right): $T_a = 297.5$ K and heater from the bottom. Best-fit parameters for Exp. 1: $k_n = 106.11$ and $k_\tau = 3.14$. Best-fit parameters for Exp. 2: $k_n = 155.55$ and $k_\tau = 6.67$	40
3.7	Experimental observations and model predictions of growth rate (top) and frequency (bottom) of the oscillations. Exp. 1 (left): $T_a = 300.5$ K and heater from the top; Exp. 2 (right): $T_a = 297.5$ K and heater from the bottom. Best-fit parameters for Exp. 1: $k_n = 106.30$, $k_\tau = 4.29$ and $R = -0.9470 + 0.1191i$. Best-fit parameters for Exp. 2: $k_n = 92.04$, $k_\tau = 5.95$ and $R = -0.9679 + 0.1305i$	42
3.8	Experimental observations and model predictions of growth rate (left) and frequency (right) of the oscillations, using Mic. 1, 2, 3, 4, 5. Optimisation is not performed, and thermoacoustic parameters are fixed to: $k_n = 106.30$ and $k_\tau = 4.29$	43
3.9	Experimental observations and model predictions of growth rate (left) and frequency (right) obtained by [34]. The reflection coefficient is treated as a regression parameter	45

List of Tables

2.1	Microphone specifications	19
2.2	Location of wall and gas thermocouples for the two experiments . .	19
3.1	Minimum, maximum, and mean values of real part, imaginary part, modulus, and angle (rad.) of the reflection coefficient R using Mic. 1, 2, 3, 4, 5	38
3.2	Minimum, maximum, and mean values of real part, imaginary part, modulus, and angle (rad.) of the reflection coefficient R using Mic. 1, 2, 3, 4, 5, 6	39

Nomenclature

Roman letters

A_g	Cross-sectional area of gas [m^2], $A_g = \pi d^2/4$
A_s	Cross-sectional area of solid wall [m^2], $A_s = \pi r(d + r)$
c	Speed of propagation (sound) [m/s]
$c_{p,g}$	Specific heat capacity of gas at constant pressure [$J kg^{-1}K^{-1}$]
c_s	Specific heat capacity of solid [$J kg^{-1}K^{-1}$]
d	Tube diameter [m]
d_w	Wire diameter of the heater [m]
h_i	Heat transfer convective coefficient from gas to solid [$Wm^{-2}K^{-1}$]
h_o	Heat transfer convective coefficient from solid to ambient [$Wm^{-2}K^{-1}$]
L	Tube length [m]
\dot{m}	Mass flow rate of gas [$kg s^{-1}$], $\dot{m} = \rho_1 A_g U_1 = \rho_g A_g U_g$
n	Interaction index [W/m^3]
Nu_i	Nusselt number for heat transfer from gas to solid, $Nu_i = h_i d / \lambda_g$
Nu_o	Nusselt number for heat transfer from solid to ambient, $Nu_o = h_o L / \lambda_a$
p	Gas pressure [Pa]
\dot{Q}	Thermal power [W]
q/\dot{Q}'_h	Thermal power released by the heater per unit length [W/m]
\dot{Q}_i	Thermal power from gas to solid [W], $\dot{Q}_i = h_i \pi_i (T_g - T_s) \delta x$
\dot{Q}_o	Thermal power from solid to ambient [W], $\dot{Q}_o = h_o \pi_o (T_s - T_a) \delta x$
R	Reflection coefficient
r	Tube thickness [m]
s	Eigenvalue [s^{-1}]

T	Temperature [K]
t	Time [s]
u/U	Gas velocity [m/s]
x	x -coordinate [m]

Greek letters

γ	Gas heat capacity ratio, $\gamma = 1.4$
$\delta(x)$	One-dimensional Dirac delta [$1/m$]
δx	Length of small element of tube [m]
Θ^*	Non-dimensional temperature, $\Theta^* = (T - T_1)/T_1$
κ_i	Pressure loss coefficient
λ	Thermal conductivity [$Wm^{-1}K^{-1}$]
π_i	Perimeter inside of solid wall [m], $\pi_i = \pi d$
π_o	Perimeter outside of solid wall [m], $\pi_o = \pi(d + 2r)$
ρ	Density [$kg\ m^{-3}$]
τ	Time-delay [s]

Subscripts and superscripts

$\overline{(\cdot)}_1$	Average gas quantity upstream of the heater
$\overline{(\cdot)}_2$	Average gas quantity downstream of the heater
$(\cdot)_1$	Gas quantity evaluated at the tube inlet section
$(\cdot)_2$	Gas quantity evaluated at the tube outlet section
$(\cdot)_a$	Quantity that refers to gas at ambient conditions
$(\cdot)_d$	Quantity downstream of the heater
$(\cdot)_g$	Quantity that refers to gas
$(\cdot)_H$	Quantity that refers to gas, evaluated at the heater location
$(\cdot)_s$	Quantity that refers to solid wall
$(\cdot)_u$	Quantity upstream of the heater
$(\cdot)^*$	Non-dimensional quantity
$(\cdot)'$	First-order perturbed quantity

Chapter 1

Introduction

Thermoacoustics is a research area that lies at the boundary between combustion, fluid dynamics, and acoustics. In simple terms, thermoacoustic instabilities occur when heat release rate unsteadiness (for example, but not necessarily, caused by a combustion process) interacts in a constructive way with the acoustic field, amplifying noise. These oscillations may cause structural damage and component failure, hence severe problems in the design phase of aero and space engines. This provides numerous research groups around the world with a strong motivation to study such phenomena, which are still far from being completely understood.

The first scientific realisation of the existence of a thermoacoustic instability dates back to 1802, when Higgins [1] observed generation of sound upon placing a hydrogen flame into tubes of different materials. A qualitative explanation of this phenomenon arrived more than seventy years later, when Lord Rayleigh published a famous paper [2] in which he realised that “vibration is encouraged” when providing heat to a gas during moments of high pressure, and when taking out heat from a gas during moments of low pressure. Indeed, the driving mechanism of a thermoacoustic instability is similar to the one that drives a piston engine, where a gas is first compressed and then burned at high pressure, and a piston expands producing mechanical work, which is larger than the compression work, thus leading to a net conversion of thermal to mechanical energy. In a thermoacoustic analogy, the piston can be seen as an acoustic wave that periodically perturbs the flame, which replaces the hot gas. If this perturbation is such that more heat is released during moments of high pressure, as occurs in a piston engine, then there is more available work which, if not dissipated, contributes to increase the oscillation amplitude, thus leading to instability. This criterion, known as the Rayleigh criterion, was mathematically formulated in 1954 by Putnam and Dennis [3] as

$$\oint_{\Delta t} \int_V q'(\mathbf{x}, t) p'(\mathbf{x}, t) d\mathbf{x} dt > 0 \quad (1.1)$$

where $q'(\mathbf{x}, t)$ and $p'(\mathbf{x}, t)$ represent the fluctuating part of heat release rate and pressure, respectively, both at a given point in space \mathbf{x} and instant in time t . Note that the first is a cyclic integral to be evaluated over a time period of length Δt . Eq. (1.1) expresses a condition for instability to occur, i.e. that heat must be added sufficiently in phase with the pressure. This equation was then revisited by Chu [4] to include the contributions of viscous dissipation and acoustic radiation from

the system boundaries, whose effect is to subtract energy from the system, thus contributing to its stability.

At the industrial scale, thermoacoustic instabilities were first noted during the development of liquid-propellant rocket engines [5]. Aero and space engines are particularly susceptible to thermoacoustic oscillations due to their high powers, which mean that even a very inefficient thermoacoustic mechanism can sustain large amplitude oscillations. For example, during the development of the Saturn Rocket F-1 engine, pressure oscillations as large as the mean pressure of the combustion chamber were observed. Oefelein and Yang [6] report that a stable design of this engine required approximately 2000 full-scale tests. Indeed, thermoacoustic instabilities tend to appear in the later stages of the design process of an engine, and are rarely predicted accurately by lab-scale or component tests. On top of this, one has to consider recently introduced stricter legislations regarding NO_x emissions. This imposes a shift towards a different combustion technology, where the gas turbine operates under lean and premixed operating conditions. Engines operating at these conditions, however, are more susceptible to combustion instabilities because of their flames' very high sensitivity to equivalence ratio perturbations, system geometry, boundary conditions, and other factors, as reported by Candel [7, 8].

These observations have provided the motivation for research on how to reliably predict these phenomena. Alternative methods to trial-and-error experimental testing rely on a wide variety of computational methods. Large Eddy Simulations (LES) numerically solve the governing equations, including acoustics and chemistry, on a computational grid representing the physical domain. The way in which this is done is by resolving the large-scale turbulent structures, and modelling the subgrid-scale ones by means of appropriate turbulence models. LES is an excellent tool at predicting both the flow field and the main unstable modes, even on very complex geometries [9]. The downside of this is the still prohibitive computational cost. An alternative to LES are low-order models, a simplified approach that aims at capturing only the most relevant physics of the problem. An example are network models [10, 11], which work best for non-complex geometries. The idea of a network model is to split the domain into different elements, each of which is characterised by homogeneous properties (ρ, u, p) , and connect the different elements by means of jump conditions. The assumption that is always made, which is often reasonable, is that the flame is acoustically compact, meaning that the length of the flame is negligible compared to the length of the first acoustic mode.

These computational methods require a model of the relationship that describes how heat release fluctuations interact with acoustic (i.e. pressure and velocity) fluctuations. This model always introduces a degree of uncertainty. A frequently adopted thermoacoustic model is the so-called $n - \tau$ model, often related by people to Crocco [5], but first introduced by Summerfield [12]. According to this model, heat release fluctuations are assumed to be proportional to velocity fluctuations through an interaction index, n , and time-delayed with respect to them by a factor of τ . Despite being quite old, this model is still extensively used today, because it contains the most influential physics of thermoacoustic instabilities.

It is clear that any approach mentioned above, regardless whether experimental or computational, comes with advantages and disadvantages. In this study, we

develop a novel methodology that combines both approaches. Considering (i) the very high sensitivity of the thermoacoustic mechanism to system parameters and boundary conditions [13], and (ii) the uncertainty related to them [14], it is natural to address the problem in a probabilistic fashion. The problem of having a computational model that depends on unknown parameters with unknown uncertainties represents a classical inverse problem. Many engineering problems fall into this category, and therefore many mathematical tools have been developed for this purpose over the last decades [15]. It is possible to infer model parameters and states of the system by extracting information from either experimental observations or high-fidelity simulations. This technique is referred to as data assimilation [16], and was first developed for meteorology problems. Assimilating data into a low-order model allows, in principle, a qualitatively accurate model to be turned into a quantitatively accurate model.

In this study, this novel methodology is developed and applied to a simple thermoacoustic system: an electrically-heated Rijke tube. This system, first studied by Rijke [17] in 1859, represents a perfect device to investigate thermoacoustic oscillations because, in spite of its simplicity, it contains much of the relevant physics present in a real engine. The goal of the project is therefore to derive and train a physics-based model of the system to correctly predict thermoacoustic oscillations, inferring its unknown parameters and related uncertainties from many thousand experimental measurements. The advantage of such a formulation, with respect to an entirely data-driven approach, is the gain in flexibility. Indeed, a physics-based model should remain, at least in principle, reasonably accurate when extrapolating beyond the range tested, due to the fact that it is based on physical principles. Although not being addressed in this study, it is worth mentioning that the high sensitivity of a thermoacoustic system to operating parameters and boundary conditions motivates the use of adjoint methods and gradient-based optimization to work out the small changes to make to the system in order to stabilise it [18], once the physics-based model is validated.

Chapter 2

Background theory

The aim of this chapter is to provide a detailed description of the theory that is at the base of this work. We first derive in Section 2.1 a base flow model able to predict the buoyancy-driven flow that sets up in the Rijke tube once the heater is switched on. Because this model contains unknown parameters, we infer these by assimilating experimental data into the base flow model every time they become available. For this, an Ensemble Kalman Filter is used (Section 2.2), which provides an optimal estimate and the related uncertainty for state and parameters, based on the uncertainties of both the experiments and the model predictions. As will be shown at the end of this study, a good knowledge of the base flow is necessary in order to accurately predict thermoacoustic oscillations. A thorough description of the experimental setup and data processing is provided in Sections 2.3 and 2.4. We then model in Section 2.5 the acoustics inside the duct using a network model, following the approach of [10, 11]. To do this, we use the linearised acoustics equations with the related jump conditions in the assumption of no mean flow, because the Mach number inside the duct is negligibly small. A classic $n - \tau$ thermoacoustic model is adopted to relate heat release fluctuations to velocity fluctuations. The thermoacoustic model parameters are estimated using simple regression, as explained in Section 2.6.

2.1 Base flow model

2.1.1 Model

We consider an infinitesimal element of tube of length δx . The energy balance in the solid reads

$$(\rho_s A_s \delta x) c_s \frac{\partial T_s}{\partial t} = -\frac{\partial \dot{Q}_s}{\partial x} \delta x - \dot{Q}_o + \dot{Q}_i \quad (2.1)$$

By using Newton's cooling law and Fourier's law, and by assuming constant thermal conductivity, Eq. (2.1) can be re-written as

$$\rho_s A_s c_s \frac{\partial T_s}{\partial t} = \lambda_s A_s \frac{\partial^2 T_s}{\partial x^2} - h_o \pi_o (T_s - T_a) + h_i \pi_i (T_g - T_s) \quad (2.2)$$

By re-arranging and replacing the heat transfer convective coefficients with the Nusselt numbers, one obtains

$$\frac{\partial T_s}{\partial t} = \frac{\lambda_s}{\rho_s c_s} \frac{\partial^2 T_s}{\partial x^2} - Nu_o \frac{\lambda_a}{\rho_s c_s} \frac{\pi_o}{A_s L} (T_s - T_a) + Nu_i \frac{\lambda_g}{\rho_s c_s} \frac{\pi_i}{A_s d} (T_g - T_s) \quad (2.3)$$

The same procedure can be applied to an infinitesimal slug of gas moving at local speed U_g , thus giving

$$(\rho_g A_g \delta x) c_{p,g} \left(\frac{\partial T_g}{\partial t} + U_g \frac{\partial T_g}{\partial x} \right) = - \frac{\partial \dot{Q}_g}{\partial x} \delta x - \dot{Q}_i + \dot{Q}'_h \delta x \quad (2.4)$$

In Eq. (2.4) the source term \dot{Q}'_h accounts for the influence of the electric heater present in the tube. One can now apply the ideal gas law, Fourier's law, and replace the gas velocity with the mass flow rate. Furthermore, one can use the continuity equation in its integral form, $\frac{\partial(\rho U A)}{\partial x} = 0$, to obtain the following expression

$$\rho_a A_g c_{p,g} \frac{T_1}{T_g} \frac{\partial T_g}{\partial t} + \dot{m} c_{p,g} \frac{\partial T_g}{\partial x} = \lambda_g A_g \frac{\partial^2 T_g}{\partial x^2} - Nu_i \lambda_g \frac{\pi_i}{d} (T_g - T_s) + \dot{Q}'_h \quad (2.5)$$

For simplicity, in the present study the gas thermal conductivity is assumed to be constant and equal to the thermal conductivity of air at ambient conditions, i.e. $\lambda_g = \lambda_a$. Notice that radiation from the heat source to the surroundings is not modelled in order to keep the model simple.

In addition to the energy equation for solid and gas, we solve for the integral momentum equation, in which we assume (i) the unsteady term to be negligible, i.e. the mass flow rate changes very rapidly compared to the buoyancy force, and (ii) the pressure losses to be concentrated at the heater location, here denoted by the subscript H , and modelled through a pressure loss coefficient k_i . Thus, the integral momentum equation reads

$$\int_0^L (\rho_1 - \rho_g) g dx - k_i \frac{\rho_H U_H^2}{2} = \rho_2 U_2^2 - \rho_1 U_1^2 \quad (2.6)$$

By using again the continuity equation in its integral form, as well as the ideal gas law, Eq. (2.6) can be re-written as follows

$$\int_0^L (\rho_1 - \rho_g) g dx - k_i \frac{\rho_1 U_1^2}{2} \frac{T_H}{T_1} = \rho_1 U_1^2 \left[\left(\frac{A_1}{A_2} \right)^2 \frac{\rho_1}{\rho_2} - 1 \right] \quad (2.7)$$

In our case $A_1 = A_2$, but in the most general case these areas might be different due to the presence of an iris at the tube outlet such that $A_2 = \pi d_i^2/4$. If we now express the cross sectional areas in terms of the diameters, and divide both sides by the inlet density, which coincides with the ambient density to a good approximation, Eq. (2.7) becomes

$$\int_0^L \frac{(\rho_1 - \rho_g)}{\rho_1} g dx = U_1^2 \left[\frac{k_i}{2} \frac{T_H}{T_1} + \left(\frac{d}{d_i} \right)^4 \frac{T_2}{T_1} - 1 \right] \quad (2.8)$$

2.1.2 Non-dimensionalisation

It is convenient to non-dimensionalise the equations previously derived, i.e. Eqs. (2.3), (2.5) and (2.8), to simplify the analysis. To do so, we use the following reference scales, arbitrarily chosen

L	$[m]$	Tube length
g	$[ms^{-2}]$	Gravitational acceleration
T_1	$[K]$	Inlet temperature (equal to ambient temperature)

Notice that (i) this choice naturally gives rise to a time reference scale, given by $(L/g)^{1/2}$, and a velocity reference scale, given by $(gL)^{1/2}$; (ii) temperatures are non-dimensionalised in terms of temperature differences, i.e. the generic temperature T , upon non-dimensionalisation, becomes $\Theta^* = \frac{T-T_1}{T_1}$, where T_1 is the inlet temperature; (iii) non-dimensional quantities are denoted by $(\cdot)^*$ to distinguish them from the respective dimensional quantities.

The energy equation for the solid therefore becomes

$$\frac{T_1}{(L/g)^{1/2}} \frac{\partial \Theta_s^*}{\partial t^*} = \frac{\lambda_s}{\rho_s c_s} \frac{T_1}{L^2} \frac{\partial^2 \Theta_s^*}{\partial x^{*2}} - Nu_o \frac{\lambda_a}{\rho_s c_s} \frac{\pi_o}{A_s L} T_1 \Theta_s^* + Nu_i \frac{\lambda_a}{\rho_s c_s} \frac{\pi_i}{A_s d} T_1 (\Theta_g^* - \Theta_s^*) \quad (2.9)$$

We now define the following non-dimensional parameters

$$\lambda_1^* = \frac{(L/g)^{1/2}}{L^2} \frac{\lambda_s}{\rho_s c_s} = \frac{1}{L(gL)^{1/2}} \frac{\lambda_s}{\rho_s c_s} \quad (2.10)$$

$$\lambda_2^* = \frac{\pi_o (L/g)^{1/2}}{A_s L} \frac{\lambda_a}{\rho_s c_s} = \frac{\pi_o}{A_s (gL)^{1/2}} \frac{\lambda_a}{\rho_s c_s} \quad (2.11)$$

$$\lambda_3^* = \frac{\pi_i (L/g)^{1/2}}{A_s d} \frac{\lambda_a}{\rho_s c_s} = \frac{\pi_i}{A_s (gL)^{1/2}} \frac{L}{d} \frac{\lambda_a}{\rho_s c_s} \quad (2.12)$$

Therefore, Eq. (2.9) becomes

$$\frac{\partial \Theta_s^*}{\partial t^*} = \lambda_1^* \frac{\partial^2 \Theta_s^*}{\partial x^{*2}} - Nu_o \lambda_2^* \Theta_s^* + Nu_i \lambda_3^* (\Theta_g^* - \Theta_s^*) \quad (2.13)$$

By non-dimensionalising the energy equation for the gas, we obtain

$$\begin{aligned} \rho_a A_g c_{p,g} \frac{T_1}{(L/g)^{1/2}} \frac{1}{(\Theta_g^* + 1)} \frac{\partial \Theta_g^*}{\partial t^*} \\ + \rho_a A_g c_{p,g} \frac{T_1}{(L/g)^{1/2}} U_1^* \frac{\partial \Theta_g^*}{\partial x^*} = \lambda_a A_g \frac{T_1}{L^2} \frac{\partial^2 \Theta_g^*}{\partial x^{*2}} - Nu_i \lambda_a \frac{\pi_i}{d} T_1 (\Theta_g^* - \Theta_s^*) + \dot{Q}'_h \end{aligned} \quad (2.14)$$

Upon defining the following non-dimensional parameters

$$\lambda_4^* = \frac{(L/g)^{1/2}}{L^2} \frac{\lambda_a}{\rho_a c_{p,g}} = \frac{1}{L(gL)^{1/2}} \frac{\lambda_a}{\rho_a c_{p,g}} \quad (2.15)$$

$$\lambda_5^* = \frac{\pi_i (L/g)^{1/2}}{A_g d} \frac{\lambda_a}{\rho_a c_{p,g}} = \frac{\pi_i}{A_g (gL)^{1/2}} \frac{L}{d} \frac{\lambda_a}{\rho_a c_{p,g}} \quad (2.16)$$

$$\dot{Q}_h^{*'} = \frac{(L/g)^{1/2}}{T_1} \frac{\dot{Q}_h'}{\rho_a A_g c_{p,g}} \quad (2.17)$$

we can re-write the energy equation for the gas as follows

$$\frac{1}{(\Theta_g^* + 1)} \frac{\partial \Theta_g^*}{\partial t^*} + U_1^* \frac{\partial \Theta_g^*}{\partial x^*} = \lambda_4^* \frac{\partial^2 \Theta_g^*}{\partial x^{*2}} - Nu_i \lambda_5^* (\Theta_g^* - \Theta_s^*) + \dot{Q}_h^{*'} \quad (2.18)$$

Eventually, we non-dimensionalise the integral momentum equation

$$\int_0^1 \left(1 - \frac{T_1}{T_g}\right) g L dx^* = U_1^{*2} g L \left[\frac{k_i T_H}{2 T_1} + \left(\frac{d}{d_i}\right)^4 \frac{T_2}{T_1} - 1 \right] \quad (2.19)$$

$$\int_0^1 \frac{\Theta_g^*}{\Theta_g^* + 1} dx^* = U_1^{*2} \left[\frac{k_i}{2} (\Theta_H^* + 1) + \left(\frac{d}{d_i}\right)^4 (\Theta_2^* + 1) - 1 \right] \quad (2.20)$$

2.1.3 Summary of the non-dimensional model

The model consists of two energy equations, one for the solid and one for the gas, and the integral momentum equation. Upon non-dimensionalisation, these can be written as follows

$$\frac{\partial \Theta_s^*}{\partial t^*} = \lambda_1^* \frac{\partial^2 \Theta_s^*}{\partial x^{*2}} - Nu_o \lambda_2^* \Theta_s^* + Nu_i \lambda_3^* (\Theta_g^* - \Theta_s^*) \quad (2.21)$$

$$\frac{1}{(\Theta_g^* + 1)} \frac{\partial \Theta_g^*}{\partial t^*} + U_1^* \frac{\partial \Theta_g^*}{\partial x^*} = \lambda_4^* \frac{\partial^2 \Theta_g^*}{\partial x^{*2}} - Nu_i \lambda_5^* (\Theta_g^* - \Theta_s^*) + \dot{Q}_h^{*'} \quad (2.22)$$

$$\int_0^1 \frac{\Theta_g^*}{\Theta_g^* + 1} dx^* = U_1^{*2} \left[\frac{k_i}{2} (\Theta_H^* + 1) + \left(\frac{d}{d_i}\right)^4 (\Theta_2^* + 1) - 1 \right] \quad (2.23)$$

2.1.4 Numerical implementation

In the derived model, the coefficients λ^* and $\dot{Q}_h^{*'}$ are known quantities. On the other hand, the three model parameters, namely Nu_o , Nu_i and k_i , are unknown quantities, whose value will be defined and updated in a probabilistic fashion using the Ensemble Kalman Filter. Details are provided in Section 2.2. At this stage, we suppose their value to be known.

The derived model consists of two coupled PDEs, namely Eqs. (2.21) and (2.22), which can be simultaneously solved once the non-dimensional inlet velocity, U_1^* , is known. From a physical point of view, this is a typical problem of conjugate heat

transfer, i.e. a problem where the knowledge of the heat transfer to one phase (e.g. solid) is necessary to get information regarding the heat transfer to another phase (e.g. fluid), and vice versa. The non-dimensional inlet velocity can be derived using Eq. (2.23), which requires the knowledge of the gas temperature profile, Θ_g^* . This couples the three equations. From a numerical point of view, the steps are: (i) define the initial temperature profiles, Θ_g^* and Θ_s^* ; (ii) approximate numerically the left-hand side of Eq. (2.23) and solve for U_1^* , algebraically; (iii) using a suitable numerical method, solve for Θ_g^* and Θ_s^* ; (iv) starting from step (ii), and using the values of Θ_g^* and Θ_s^* computed at step (iii), iterate in a loop marching over time.

The two one-dimensional energy equations are discretized on a grid of $N = 101$ points using the finite difference method [19]. Because the Rijke tube used in this study has a length $L = 1$ m, this corresponds to placing a numerical grid point every centimeter between inlet and outlet. The finite difference method relies on replacing exact derivatives of a certain function with finite differences of the same function at discrete locations (or, more generally, with a linear combination of the function evaluated at different grid points). In particular, the approximation of the n -th derivative on a uniform grid with spacing h is given by

$$\left. \frac{\delta^n \phi}{\delta x^n} \right|_i = \frac{1}{h^n} \sum_{j=l}^r \omega_j \phi_{i+j} \quad (2.24)$$

where $l(\leq 0)$ and $r(\geq 0)$ define the stencil. The general approach to derive the weights ω_j is to expand each ϕ_{i+j} about ϕ_i using Taylor series, and derive conditions that guarantee that on the right-hand side of Eq. (2.24) the only term to appear is the one that multiplies $\frac{d^n \phi}{dx^n}$. More details are provided in [20, 21]. An intrinsic relationship between the stencil and the order of accuracy p of the n -th-order approximation exists. This can be summarised as follows

$$\begin{aligned} \text{Arbitrary stencil:} & \quad p = r - l + 1 - n \\ \text{Centred stencil, } n \text{ odd:} & \quad p = (2r + 1) - n, \quad \omega_j = -\omega_{-j}, \quad \omega_0 = 0 \\ \text{Centred stencil, } n \text{ even:} & \quad p = 2(r + 1) - n, \quad \omega_j = \omega_{-j}, \quad \omega_0 \neq 0 \end{aligned}$$

One should note that for centred stencils $r = -l$, and hence centred stencils always guarantee an extra order of accuracy for approximations of derivatives of even order, compared to biased stencils.

In addition to accuracy, one should also take into account the stability of a particular approximation. Indeed, for given approximations of derivatives, and for given governing equations, there exist a critical timestep above which numerical instabilities occur. The critical timestep can sometimes be derived analytically using several techniques (e.g. Von Neumann Stability Analysis, Modified Equation Analysis, etc.), but because of the presence of a source term in our governing equations, the analysis becomes more complicated. The simplest approach that can be used to circumvent this analysis is a trial-and-error approach, where the timestep is iteratively chosen such that instabilities do not occur.

In the present study, fourth-order accurate centred approximations are used for both first- and second-order spatial derivatives. Fourth-order accurate biased approximations are used near the boundaries. The corresponding weights are tabulated in [20] or in any book on numerical methods. For the sake of simplicity,

Neumann boundary conditions on both the solid and the gas temperatures are imposed at the boundaries. This turns out to be a reasonable assumption, provided that the grid is fine enough, as in our case. To ease the implementation, differentiation matrices are used for spatial discretization. The time integration method is the explicit fourth-order accurate Runge-Kutta, known as RK4. This guarantees higher accuracy and allows for larger timesteps, if compared with the (explicit) first-order accurate Euler method. The numerical method used to solve the integral that appears on the left-hand side of Eq. (2.23) is the second-order accurate midpoint rule, for the sake of simplicity.

The source term that appears in the energy equation for the gas is modelled with a Gaussian distribution, i.e. $\dot{Q}_h^* \propto e^{-\frac{(x^*-\mu)^2}{\sigma^2}}$, centred at the heater location ($\mu \equiv x_H^* = x_H/L = 0.25$), with variance $\sigma^2 = 0.001$, and properly normalized so as to guarantee that its integral over the length of the pipe equals the non-dimensional thermal power provided to the system, \dot{Q}_h^* . As an example, we report in Fig. 2.1 the shape of the source term when its integral is equal to 1. Notice that the choice of setting $\sigma^2 = 0.001$ is arbitrary because this is a parameter that cannot be directly measured. It is worth highlighting that the smaller the variance, the more the Gaussian shape tends to resemble the Dirac delta function. This means that, as the variance is reduced, one should make the grid finer and finer in order to avoid introducing a discontinuity in the energy equation for the gas, which the finite difference method would not be able to cope with. Therefore the value $\sigma^2 = 0.001$ is chosen as best compromise, given the grid of $N = 101$, to avoid numerical instabilities.

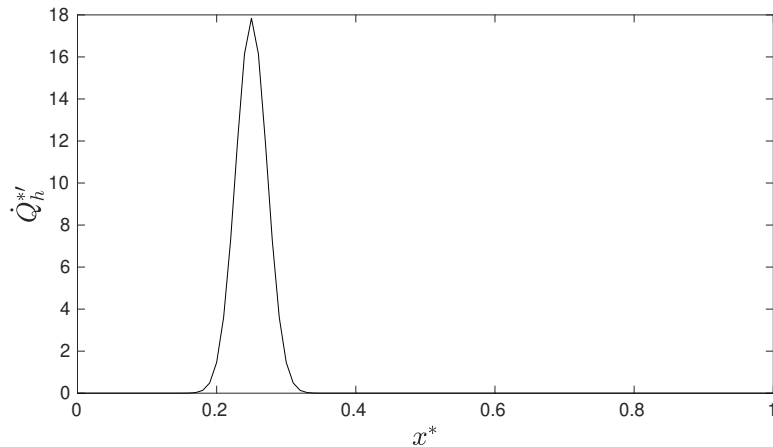


Figure 2.1: Shape of the source term normalized such that it integrates to 1. Parameters of the Gaussian distribution are $\mu = 0.25$, $\sigma^2 = 0.001$.

Notice that the right-hand side of Eq. (2.23) requires the temperatures at the heater location and at the outlet section. Because of thermal inertia, the gas in the tube reaches its maximum temperature slightly downstream of the heater location. The term Θ_H^* , which should be evaluated at the heater location, is instead evaluated where the gas temperature is maximum, for modelling purposes. It has been observed no significant change in the results when the temperature at the heater location is used.

The numerical implementation of the base flow model is reported in a Matlab[®]

routine that can be found in Appendix A, in which the values of all the known parameters of the problem, such as geometry and thermophysical properties, are also reported.

2.2 Ensemble Kalman Filter

In many engineering problems we have to deal with uncertainties both in physical models, which rely on simplifying assumptions and may contain unknown parameters, and in experiments, which are always affected by noise. Therefore it is often necessary to combine physical models with experimental observations in order to find better estimates and related uncertainties of the model solution (and, if possible, model parameters) conditioned on the measurements. This is the definition of an inverse problem, which we deal with in this study. Conversely, when the model is sufficiently reliable and the uncertainties in the parameters or in the initial/boundary conditions are known, one may want to find the way in which these uncertainties propagate through the model and affect the results. This is the definition of a forward problem.

We start the discussion by providing the reader with some basic notions of statistics. We then explain how the classic Kalman Filter (KF) works, highlight drawbacks of this technique, and eventually present the Ensemble Kalman Filter (EnKF) as a solution to these drawbacks. The discussion is based on the book of Evensen [16], who first derived the Ensemble Kalman Filter.

Given two random variables Ψ and Φ , the probability of two events, $\Psi = \psi$ and $\Phi = \phi$, occurring together is defined by the *joint* probability density function $f(\psi, \phi)$. The *conditional* probability density function $f(\psi|\phi)$ describes the probability of a certain event $\Psi = \psi$ to occur, given the event $\Phi = \phi$. The *marginal* probability density function $f(\psi)$ gives instead the probability of an event $\Psi = \psi$ to occur, regardless of Φ , and is given by

$$f(\psi) = \int_{-\infty}^{+\infty} f(\psi, \phi) d\phi \quad (2.25)$$

Moreover, we have that

$$f(\psi, \phi) = f(\psi|\phi)f(\phi) = f(\phi|\psi)f(\psi) \quad (2.26)$$

from which it is possible to derive Bayes' theorem

$$f(\psi|\phi) = \frac{f(\psi)f(\phi|\psi)}{f(\phi)} \quad (2.27)$$

If we now define $\boldsymbol{\psi}$ to be the model state vector, and \mathbf{d} to be the vector of measurements, Bayes' theorem reads

$$f(\boldsymbol{\psi}|\mathbf{d}) = \frac{f(\boldsymbol{\psi})f(\mathbf{d}|\boldsymbol{\psi})}{f(\mathbf{d})} \quad (2.28)$$

According to such theorem, the probability of the model given the data $f(\boldsymbol{\psi}|\mathbf{d})$ (often called the *posterior*) is proportional to the product between the marginal

probability of the model $f(\boldsymbol{\psi})$ (often called the *prior*), that is the probability of the model with no knowledge of the data, and the probability of the data given the model $f(\mathbf{d}|\boldsymbol{\psi})$ (often called the *likelihood* function). Notice that the denominator is just the integral of the numerator because of Eqs. (2.25) and (2.26), hence the denominator normalises the numerator so that the right-hand side of Eq. (2.28) integrates to one. This theorem represents the fundamentals of the following discussion.

The problem is now how to combine a model prediction of a state variable at a given time with a set of experimental realisations available at this particular time. The main assumption behind the classic Kalman Filter is that the error statistics (i.e. the covariance matrix) of both the model prediction and the measurements are known. To ease comprehension, it is worth starting from the scalar case. In particular, we assume that the true state of a certain variable at a certain time, ψ^t , can be expressed in the following two ways:

$$\psi^f = \psi^t + p^f \quad (2.29)$$

$$d = \psi^t + \varepsilon \quad (2.30)$$

where ψ^f is a model forecast, d is a measurement of the variable, p^f is the unknown error in the forecast and ε is the unknown error in the measurement. Finding a better estimate ψ^a of ψ^t requires additional information about the error terms, therefore we make the following assumptions:

$$\overline{p^f} = 0, \quad \overline{\varepsilon} = 0, \quad \overline{\varepsilon p^f} = 0, \quad \overline{(p^f)^2} = C_{\psi\psi}^f, \quad \overline{(\varepsilon)^2} = C_{\varepsilon\varepsilon} \quad (2.31)$$

where the overbar indicates ensemble averaging or expected value. We now seek for an unbiased linear estimator

$$\psi^a = \psi^t + p^a = \alpha_1 \psi^f + \alpha_2 d \quad (2.32)$$

where p^a is the unknown error in the analysed estimate, and we define

$$\overline{p^a} = 0, \quad \overline{(p^a)^2} = C_{\psi\psi}^a \quad (2.33)$$

By substituting Eqs. (2.29) and (2.30) into Eq. (2.32) we get

$$\psi^t + p^a = \alpha_1(\psi^t + p^f) + \alpha_2(\psi^t + \varepsilon) \quad (2.34)$$

We determine the coefficient α_1 by imposing $\overline{p^a} = 0$. By doing so, the expected value of Eq. (2.34) becomes

$$\overline{\psi^t} = \alpha_1 \overline{\psi^t} + \alpha_2 \overline{\psi^t} = (\alpha_1 + \alpha_2) \overline{\psi^t} \quad (2.35)$$

therefore we require $\alpha_1 = 1 - \alpha_2$ and Eq. (2.32) becomes

$$\psi^a = \psi^t + p^a = (1 - \alpha_2)\psi^f + \alpha_2 d = \psi^f + \alpha_2(d - \psi^f) \quad (2.36)$$

If we now use Eqs. (2.29) and (2.30), this equation can be re-written as

$$p^a = p^f + \alpha_2(\varepsilon - p^f) \quad (2.37)$$

The expected value of the left-hand side's square of such equation gives the error variance of the analysed estimate

$$\begin{aligned}\overline{(p^a)^2} &= C_{\psi\psi}^a \\ &= \overline{(p^f)^2} + 2\alpha_2 \overline{p^f(\varepsilon - p^f)} + \alpha_2^2 \overline{\varepsilon^2 - 2\varepsilon p^f + (p^f)^2} \\ &= C_{\psi\psi}^f - 2\alpha_2 C_{\psi\psi}^f + \alpha_2^2 (C_{\varepsilon\varepsilon} + C_{\psi\psi}^f)\end{aligned}\quad (2.38)$$

We determine the coefficient α_2 by minimising the error variance $C_{\psi\psi}^a$

$$\frac{dC_{\psi\psi}^a}{d\alpha_2} = -2C_{\psi\psi}^f + 2\alpha_2(C_{\varepsilon\varepsilon} + C_{\psi\psi}^f) = 0 \quad (2.39)$$

which gives

$$\alpha_2 = \frac{C_{\psi\psi}^f}{C_{\varepsilon\varepsilon} + C_{\psi\psi}^f} \quad (2.40)$$

Thus, in conclusion, if the error statistics of both the model prediction and the experimental measurement are known, it is possible to provide a better estimate of the state by combining these two pieces of information. By requiring the estimator to be unbiased and with minimum error variance, the analysed estimate and its variance become

$$\psi^a = \psi^f + \frac{C_{\psi\psi}^f}{C_{\varepsilon\varepsilon} + C_{\psi\psi}^f}(d - \psi^f), \quad C_{\psi\psi}^a = C_{\psi\psi}^f \left(1 - \frac{C_{\psi\psi}^f}{C_{\varepsilon\varepsilon} + C_{\psi\psi}^f}\right) \quad (2.41)$$

Intuitively, this result is connected to Bayes' theorem. In fact, it is a very special case of the theorem, which occurs when Gaussian priors are assumed. Indeed, by supposing

$$f(\psi) \propto \exp\left(-\frac{1}{2}(\psi - \psi^f)(C_{\psi\psi}^f)^{-1}(\psi - \psi^f)\right) \quad (2.42)$$

and

$$f(d|\psi) \propto \exp\left(-\frac{1}{2}(\psi - d)C_{\varepsilon\varepsilon}^{-1}(\psi - d)\right) \quad (2.43)$$

and by recalling Eq. (2.28), the posterior pdf can be written as

$$f(\psi|d) \propto \exp\left(-\frac{1}{2}\chi[\psi]\right) \quad (2.44)$$

where

$$\chi[\psi] = (\psi - \psi^f)(C_{\psi\psi}^f)^{-1}(\psi - \psi^f) + (\psi - d)C_{\varepsilon\varepsilon}^{-1}(\psi - d) \quad (2.45)$$

The least squares solution ψ^a that gives a minimum for $\chi[\psi]$, also gives a maximum of $f(\psi|d)$, i.e. the maximum likelihood estimate. This will always be true as long

as $f(\psi)$ and $f(d|\psi)$ are Gaussian distributed. The minimum of $\chi[\psi]$ is found by setting

$$\frac{d\chi}{d\psi} = 2(\psi - \psi^f)(C_{\psi\psi}^f)^{-1} + 2(\psi - d)C_{\varepsilon\varepsilon}^{-1} = 0 \quad (2.46)$$

Solving for ψ gives the result obtained in Eq. (2.41), which proves that, the minimum variance estimate (analogous to the least square approach) coincides with the maximum likelihood estimate in the case with Gaussian priors.

The result obtained in Eq. (2.41) can be extended to multiple dimensions. In this case, we write Eqs. (2.29) and (2.30) as

$$\boldsymbol{\psi}^f = \boldsymbol{\psi}^t + \mathbf{p}^f \quad (2.47)$$

$$\mathbf{d} = \mathbf{M}\boldsymbol{\psi}^t + \boldsymbol{\varepsilon} \quad (2.48)$$

where \mathbf{M} is a mapping matrix, also called measurement matrix. If the state vector is made up of n variables, and if the number of measurements is m , then the mapping matrix has dimensions $m \times n$ and in the simplest case each of its rows consists of zeros everywhere except at the measurement location, where a 1 is instead present. Similarly to Eq. (2.31), the assumptions are now

$$\overline{\mathbf{p}^f} = \mathbf{0}, \quad \overline{\boldsymbol{\varepsilon}} = \mathbf{0}, \quad \overline{\mathbf{p}^f \boldsymbol{\varepsilon}^T} = \mathbf{0}, \quad \overline{\mathbf{p}^f (\mathbf{p}^f)^T} = \mathbf{C}_{\psi\psi}^f, \quad \overline{\boldsymbol{\varepsilon} \boldsymbol{\varepsilon}^T} = \mathbf{C}_{\varepsilon\varepsilon} \quad (2.49)$$

and by using the same procedure shown above one can obtain

$$\boldsymbol{\psi}^a = \boldsymbol{\psi}^f + \mathbf{K}(\mathbf{d} - \mathbf{M}\boldsymbol{\psi}^f) \quad (2.50)$$

$$\mathbf{C}_{\psi\psi}^a = (\mathbf{I} - \mathbf{K}\mathbf{M})\mathbf{C}_{\psi\psi}^f \quad (2.51)$$

$$\mathbf{K} = \mathbf{C}_{\psi\psi}^f \mathbf{M}^T (\mathbf{M}\mathbf{C}_{\psi\psi}^f \mathbf{M}^T + \mathbf{C}_{\varepsilon\varepsilon})^{-1} \quad (2.52)$$

where the matrix \mathbf{K} is often called the Kalman gain. The equations just derived represent the analysis equations for the Kalman Filter.

So far we have considered how to compute at a given time the best conditional estimate given a prior estimate and measurements of the state, both with known uncertainties. It is now interesting to see how to adapt this concept to time dependent problems. This is called sequential data assimilation. For linear dynamics and Gaussian error, the optimal sequential data assimilation method is the Kalman Filter [16]. In the Kalman Filter, an additional equation for the covariance matrix is integrated forward in time to predict error statistics for the model forecast. The error statistics are then used to calculate a variance minimising estimate every time experimental observations become available, using the procedure shown in Eqs. (2.50) to (2.52). We first derive this additional equation for the scalar case, and then we apply the same procedure to the multi-dimensional one.

We can write a discrete dynamical model, characterised by the linear model operator G , as

$$\psi^t(t_k) = G\psi^t(t_{k-1}) + q(t_{k-1}) \quad (2.53)$$

$$\psi^t(t_0) = \Psi_0 + a \quad (2.54)$$

where q is the model error over one time step and Ψ_0 is an initial condition characterised by error a . Because the model error is unknown, the evolution of the model will be

$$\psi^f(t_k) = G\psi^a(t_{k-1}) \quad (2.55)$$

$$\psi^a(t_0) = \Psi_0 \quad (2.56)$$

which means that a forecast at time t is simply obtained by time evolving the best estimate at time t_{k-1} . This is an approximation because we do not know how the associated error evolves, but it is the best we can do. Moreover, by subtracting Eq. (2.55) from Eq. (2.53) we obtain

$$\psi_k^t - \psi_k^f = G(\psi_{k-1}^t - \psi_{k-1}^a) + q_{k-1} \quad (2.57)$$

where we have defined $\psi_k = \psi(t_k)$ and $q_k = q(t_k)$. The error covariance matrix for the forecast at time t_k is

$$\begin{aligned} C_{\psi\psi}^f(t_k) &= \overline{(\psi_k^t - \psi_k^f)^2} \\ &= G^2 \overline{(\psi_{k-1}^t - \psi_{k-1}^a)^2} + \overline{q_{k-1}^2} + 2G \overline{(\psi_{k-1}^t - \psi_{k-1}^a)q_{k-1}} \\ &= G^2 C_{\psi\psi}^a(t_{k-1}) + C_{qq}(t_{k-1}) \end{aligned} \quad (2.58)$$

where

$$\begin{aligned} C_{\psi\psi}^a(t_{k-1}) &= \overline{(\psi_{k-1}^t - \psi_{k-1}^a)^2} \\ C_{qq}(t_{k-1}) &= \overline{q_{k-1}^2} \\ C_{\psi\psi}(t_0) &= C_{aa} = \overline{a^2} \end{aligned} \quad (2.59)$$

are the model state error covariance, the model error covariance, and the initial error covariance, respectively. Notice that it has been implicitly assumed that there are no correlations between the state error, $(\psi_{k-1}^t - \psi_{k-1}^a)$, and the model error, q_{k-1} . An important remark is that the linearity hypothesis allowed us to take the operator G out of the expected value $\overline{(\cdot)}$. Without this assumption, it would not be possible to obtain the same result as that in Eq. (2.58). This is the reason why the Kalman Filter works only for linear dynamics.

In multiple dimensions, the same considerations hold, and the analogous equations to Eqs. (2.55) and (2.58) read

$$\boldsymbol{\psi}_k^f = \mathbf{G}\boldsymbol{\psi}_{k-1}^a \quad (2.60)$$

$$\mathbf{C}_{\psi\psi}^f(t_k) = \mathbf{G}\mathbf{C}_{\psi\psi}^a(t_{k-1})\mathbf{G}^T + \mathbf{C}_{qq}(t_{k-1}) \quad (2.61)$$

Therefore, the model evolves according to Eq. (2.60), the error covariance evolves according to Eq. (2.61), and every time experimental observations are available, an analysed state estimate can be calculated using Eqs. (2.50) to (2.52). When

no observations are available, we just set $\psi_k^a = \psi_k^f$ and $C_{\psi\psi}^a(t_k) = C_{\psi\psi}^f(t_k)$, and continue the integration. These equations define the Kalman Filter, which is based on the assumptions of (i) linear dynamics, (ii) Gaussian and (iii) unbiased priors.

Despite the Kalman Filter being the best method for sequential data assimilation, its use is restricted to a relatively small category of problems. Indeed, its main drawbacks are the linearity assumption, which is often not valid in practical problems, and the fact that a covariance matrix $C_{\psi\psi}^a$ needs to be stored. In particular, if the model state vector consists of n variables, the covariance matrix is made up of n^2 entries, and for problems with many thousand degrees of freedom the computational requirements associated to its storage are too high. This motivated people to find an alternative to circumvent these two problems. A new method, the Extended Kalman Filter (EKF), was derived to tackle the first drawback by linearising the nonlinear model operator in the true state $G(\psi_k^t)$ about the analysed state $G(\psi_k^a)$ using a Taylor series expansion. In other words, unlike the classic Kalman Filter, which linearly advances forward in time the mean and the covariance of a Gaussian distribution using Eqs. (2.60) and (2.61), the Extended Kalman Filter aims to propagate the same quantities allowing for weak nonlinearities. Nonetheless, this method is not very successful because for strong nonlinearities one would need to retain many high-order terms in the Taylor series expansion in order for the method to be accurate, making it cumbersome. Moreover, the EKF does not solve the problem related to the need to store a covariance matrix.

A valid alternative to the two previous methods is represented by the Ensemble Kalman Filter (EnKF). In essence, the EnKF advances forward in time sampled *members* of a prior distribution, according to the model dynamics that might be nonlinear. The members constitute the ensemble. This leads to non-Gaussian distributions of the model forecasts if the model is nonlinear. Regarding the observations, they are treated as random variables having Gaussian distribution, similarly to those of the KF and EKF. In the analysis step, forecasts and observations are combined in a similar way to that of the other methods, but the difference lies in the fact that the model forecasts are now not necessarily Gaussian distributed. If the ensemble size is sufficiently large, one could reconstruct the corresponding model forecast pdf and directly apply Bayes' theorem by taking the product between the reconstructed model forecast pdf and the measurement pdf. However, if the model state variables are many, reconstructing a pdf for each variable becomes too expensive. Thus, what is done in practice is to compute sample mean and sample covariance matrices from the ensemble members, treat each model forecast pdf as if it were Gaussian, and perform the analysis step in a similar way to that in Eqs. (2.50) to (2.52). Although the forecast distributions are treated as Gaussian when performing the analysis step, and despite the fact that the observations are assumed to be Gaussian distributed, the method takes into account the nonlinearity in the propagation step of the ensemble members. In other words, although the EnKF still relies on some simplifying assumptions, these are much less restrictive than the assumptions on which both the KF and the EKF are based. In practice, the EnKF represents a powerful tool that works well on both linear and nonlinear problems [16]. The analysis equations for each ensemble member j are very similar

to Eqs. (2.50) to (2.52) and read

$$\boldsymbol{\psi}_j^a = \boldsymbol{\psi}_j^f + \mathbf{K}_e(\mathbf{d}_j - \mathbf{M}\boldsymbol{\psi}_j^f) \quad (2.62)$$

$$\mathbf{C}_{\psi\psi}^a = (\mathbf{I} - \mathbf{K}_e\mathbf{M})(\mathbf{C}_{\psi\psi}^e)^f \quad (2.63)$$

$$\mathbf{K}^e = (\mathbf{C}_{\psi\psi}^e)^f \mathbf{M}^T (\mathbf{M}(\mathbf{C}_{\psi\psi}^e)^f \mathbf{M}^T + \mathbf{C}_{\varepsilon\varepsilon}^e)^{-1} \quad (2.64)$$

where the superscript e denotes quantities computed from the ensemble.

The second advantage of the ensemble formulation is that it speeds up the computational time because storing covariance matrices is not required. Inspection of Eqs. (2.62) to (2.64) does not allow us to appreciate this feature, since the matrices present in these equations have the same dimensions as those used in the classic Kalman Filter. However, thanks to the ensemble formulation, it is possible to manipulate these equations in a way such that the computation reduces to only vector multiplications. The method is not reported here, but the interested reader is referred to Wikipedia, which reports a quite straightforward derivation. Thus, in conclusion, the EnKF proves to be a very useful and efficient tool for sequential data assimilation, which works well for both linear and nonlinear dynamics.

In the present study, the ensemble consists of 30 members. This means that 30 simulations of the base flow with slightly different parameters and initial conditions run in parallel, and state and parameters are updated every time experimental observations become available, hence every 8 seconds. The implementation of the Ensemble Kalman Filter for the present study is by H. Yu (University of Cambridge) and is reported in Appendix B as a Python routine.

2.3 Experimental setup and data acquisition

2.3.1 Apparatus

The rig consists of a 1 m long stainless steel vertical tube with an internal diameter of 47.4 mm and a wall thickness of 1.7 mm. An electric heater is attached to two rods and held in place at $x/L = 0.25$, which corresponds to the optimal position for self-excited thermoacoustic oscillations (Saito [22]). The heater is powered by a 640 Watt EA Elektro-Automatik EA-PSI 5080-20 A DC programmable power supply controlled through National Instruments LabVIEW. The experimental apparatus, as described so far, is the same as the one used in [23, 24, 25, 26].

Six G.R.A.S. 40SA probe microphones are used to record the pressure near the inner surface of the tube at different axial locations (see Fig. 2.2). Note that, unlike the sketch in Fig. 2.2, the microphones are all placed at the same azimuthal location, because the acoustics inside the duct is assumed to be 1D. Each microphone is equipped with a 20 mm long probe, on top of which a temperature shield is mounted to protect the main body from the high temperatures experienced within the tube. Information regarding serial number, axial location, and sensitivity of each microphone are summarised in Table 2.1. The raw pressure signal is sampled at 1250 Hz, a much higher frequency than the anticipated frequencies of the thermoacoustic oscillations, 170 - 190 Hz. All data is acquired through a National Instruments

BNC-2110 DAQ device using LabVIEW. A $4\ \Omega$ Visaton FRS 8 loudspeaker is fixed at the base of the tube, as depicted in Fig. 2.2. The loudspeaker is connected to a 600 W Stage Line STA-500 Pro Power amplifier and provided an acoustic pulse controlled through National Instruments LabVIEW.

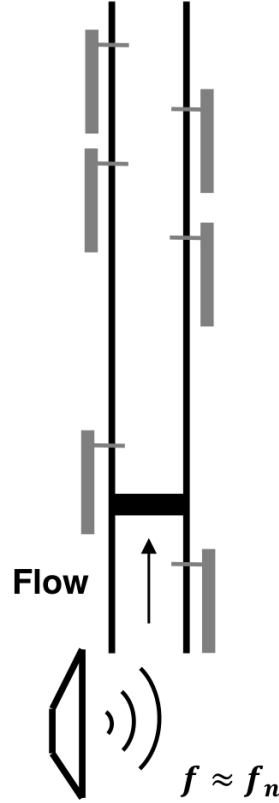


Figure 2.2: Sketch of the experimental setup. An electric heater is placed at one quarter of the tube length from the bottom. Six microphones are distributed along the tube, and a loudspeaker is placed at the bottom. Thermocouples, not shown here, are installed inside and on the external surface of the tube at different locations.

Type-K thermocouples are secured along the external surface of the tube to record the solid temperature, and at some points along the centreline of the tube. The heater is attached to two rods that hold it in place. Two types of experiments have been carried out: in both cases the heater is placed at $x/L = 0.25$, but in the first experiment the heater is inserted from the top end, whereas in the second experiment it is inserted from the bottom end. The difference lies in the length of the two rods. In particular, in the first case, the length of the rods in the tube is three times greater than in the second case. This has a strong influence on the acoustics, because it changes the amount of acoustic damping, as will be shown in Section 3.2. The exact position at which the thermocouples are located is reported in Table 2.2. Two additional thermocouples are placed (i) near the inlet section to record the ambient temperature and (ii) on the main body of the most thermally exposed microphone, i.e. Mic n° 5 in Table 2.1, to monitor its temperature. All thermocouples are logged with four TC-08 USB DAQ boxes - two of these are

Mic number	Serial number	Distance from bottom end [m]	Sensitivity [mv/Pa]
1	263705	0.95	2.99
2	263706	0.85	2.95
3	263707	0.75	2.85
4	263703	0.65	3.02
5	263702	0.35	2.80
6	263704	0.15	3.05

Table 2.1: Microphone specifications

Omega branded and two are Pico branded.

An important point is that, because we are using a 1D model, the gas temperature of the model is the bulk temperature, which differs from the centreline temperature. A 2D model would be required, however, to cope with this. We therefore make the questionable assumption that the centreline temperature does not significantly differ from the bulk temperature.

Description	Heater from the top									
x/L (Wall)	0.28	0.31	0.33	0.35	0.38	0.41	0.44	0.47	0.50	
x/L (Gas)	0.53	0.56	0.60	0.64	0.68	0.73	0.78	0.83	0.89	0.95
Description	Heater from the bottom									
x/L (Wall)	0.25	0.27	0.30	0.34	0.37	0.40	0.44	0.48		
x/L (Gas)	0.52	0.57	0.62	0.67	0.72	0.78	0.84	0.90	0.96	
x/L (Gas)	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90		

Table 2.2: Location of wall and gas thermocouples for the two experiments

2.3.2 Data acquisition

The experiment is completely automated through National Instruments LabVIEW. The LabVIEW program reads a text file where the heater powers are input as a column vector, and executes a for-loop for every line of the file, i.e. for every input power. The for-loop has a duration of 68 s, during which the following commands are executed simultaneously: (i) The desired power is provided to the system. Because voltage and current provided by the power supply fluctuate over time, a PID control is implemented so that the provided power is as close as possible to the desired power. This results in fluctuations of 1 - 2 W, which is small compared to the desired powers. (ii) After approximately 1.25 s, a 50 ms long sinusoidal signal at a frequency of 170 Hz is provided to the system. (This frequency is kept constant for all the experiments, and is very close to the natural frequency of the tube, which depends on the temperature inside the tube, and therefore slightly varies over time.) (iii) During the first 64 s, the six microphones acquire the pressure signal at a

frequency of 1250 Hz. This means that for each microphone an output file containing 80000 values is written every cycle. The remaining 4 s of the loop are used by the LabVIEW program to write output files, therefore no data is acquired during this time. (iv) During the first 64 s, the thermocouples record the temperature at a frequency of 0.125 Hz. This means that the temperature is sampled every 8 seconds, and therefore for each thermocouple an output file containing 8 values is written every cycle. As before, the remaining 4 s of the loop are employed for data writing.

The input power is varied from 10 W up to 180 W, with steps of 10 W. In the text file that LabVIEW reads, each power is input 60 times. This means that the text file has 1080 entries, and therefore the duration of the entire experiment is exactly 20.4 h (obtained from $68 \frac{s}{loop} \cdot 60 \frac{loop}{power} \cdot \frac{18 power}{3600 \frac{s}{h}}$). Because at these operating conditions the system is stable, when the system is pulsed, oscillations die out. The goal of providing a pulse is twofold: (i) extract from the pressure signal the oscillations' decay rate and the related frequency, and (ii) compute the reflection coefficient at the inlet and outlet sections by means of the multi-microphone method. This is described in detail in Section 2.4.

Thus, at the end of the experiment, one can (i) use 8640 temperature data (for each thermocouple) to assimilate them into the base flow model; (ii) obtain 1080 experimental observations of decay rate and related frequency of the oscillations; (iii) compute 1080 reflection coefficients that will feed into the thermoacoustic model - this is described in detail in Section 2.5.

2.4 Data processing

2.4.1 Experimental decay rate and frequency

In the linearly stable regime, any small perturbation dies out. The same applies to the 50 ms long pulse provided to the system every 68 s. A plot of the pressure signal recorded by a microphone is shown in Fig. 2.3a. Note that the signal amplitude depends on the axial position, therefore each microphone records a signal with a different amplitude. In particular, the first acoustic pressure mode inside the duct, given the open-open boundary condition, has ideally a pressure node at the inlet and outlet sections, and an antinode in the middle, i.e. at $x/L = 0.5$. This means that microphones placed towards the ends will record lower amplitude signals than those placed towards the centre of the tube. As a side note, it is worth mentioning that in reality the pressure nodes are located just before the inlet section and just after the outlet section because of acoustic radiation through the boundaries - the first mathematical formulation of this effect is due to Levine and Schwinger [27].

To extract the decay rate and frequency, the signal of each microphone is processed in the following way. First, a Butterworth filter, centred at the excitation frequency (170 Hz) and with size ± 25 Hz, is applied in order to filter out all the undesired frequencies. Then a Hilbert transform is applied to obtain the instantaneous amplitude, $A(t)$, and phase, $\phi(t)$, of the pressure signal, similarly to Schumm et al. [28]. From the obtained amplitude, when plotted against time and on a log-scale, it is possible to identify a linear region, corresponding to an

exponential decay. The slope of this corresponds to the decay rate of the oscillations. It is therefore enough to identify the linear region and fit a first-order polynomial to extract the decay rate (see Fig. 2.3b). The associated frequency can be extracted directly from the phase of the Hilbert transform, but in this study a Fourier transform is instead used. Both the Hilbert and the Fourier transforms are inbuilt functions in Matlab[®]. The routine that computes the decay rate and frequency of the oscillations, given the raw signal, can be found in Appendix C. Once these quantities are available for each microphone, the values are averaged over the six microphones, so that a more robust estimate is obtained. It is worth mentioning that the averaged decay rate and frequency represent the real and imaginary part of the experimental eigenvalue, s_{exp} , respectively. More details about this are provided in Section 2.5.

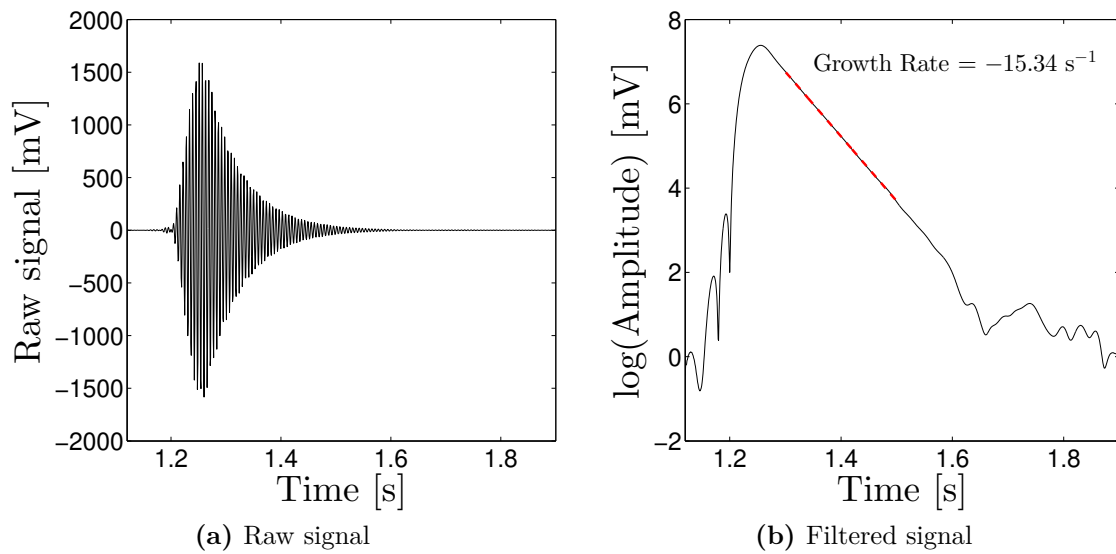


Figure 2.3: (a) 50 ms long sinusoidal signal provided to the system; (b) filtered signal: exponential decay with best-fit line in red

2.4.2 Multi-microphone method

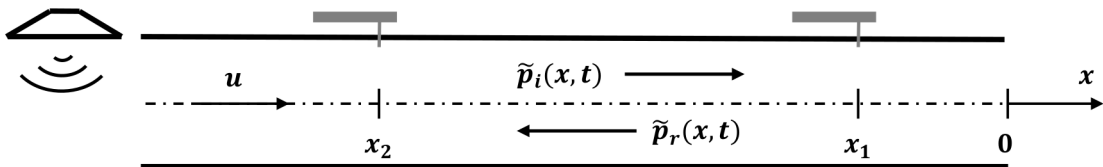


Figure 2.4: Tube of length L open at both ends with unknown reflection coefficient R at $x = 0$, excited by an acoustic pulse at $x = -L$.

The multi-microphone method (MMM) represents a more accurate and robust version of the two-microphone method (TMM), first formulated by Seybert and Ross [29]. This method computes the reflection coefficient of an acoustic boundary, given

the pressure signal measured at two (TMM) or more (MMM) different locations inside the acoustic domain. Fig. 2.4 shows a rigid tube of length L excited at $x = -L$ by an acoustic pulse, or other acoustic source, and terminated at $x = 0$ by an acoustic boundary with unknown reflection coefficient R . Mean flow is included, where u is the velocity of the fluid. The acoustic source is assumed not to vary in space. The origin of the coordinate system is at the opposite end of the tube, which simplifies the analysis. Two microphones record the pressure inside the tube at locations $x = x_1$ and $x = x_2$, both of which have a negative value due to our choice of the coordinate system. At each axial location and instant in time, the pressure $\tilde{p}(x, t)$ can be obtained by summing up the incident and reflected pressure waves [30], denoted by $\tilde{p}_i(x, t)$ and $\tilde{p}_r(x, t)$, respectively. In a narrow bandwidth, centred at frequency f , which corresponds to the excitation frequency, random motion can be regarded as harmonic motion at frequency f with a randomly varying amplitude. For plane-wave propagation (i.e. 1D acoustics) the incident and reflected waves are given by

$$\tilde{p}_i(x, t) = \tilde{a}(t)e^{i(\omega t - \kappa_i x)} \quad (2.65)$$

$$\tilde{p}_r(x, t) = \tilde{b}(t)e^{i(\omega t + \kappa_r x)} \quad (2.66)$$

where $\omega = 2\pi f$ is the angular velocity, whereas $\kappa_i = \omega/(c_0 + u)$ and $\kappa_r = \omega/(c_0 - u)$ are the incident and reflected wavenumbers, respectively, with c_0 being the speed of propagation with zero flow. Here \tilde{a} and \tilde{b} are random variables that do not depend on space since the system is assumed to be nondispersive. This means that the wave shape remains the same during propagation. The Fourier transform of the pressure signal at each location along the tube is given by

$$\tilde{P}(x, f) = \tilde{p}(x, t)e^{-i\omega t} \quad (2.67)$$

Therefore, at locations $x = x_1$ and $x = x_2$, where the pressure signal and hence its Fourier transform is known, we can write

$$\begin{cases} \tilde{P}(x_1, f) = \tilde{A}(f)e^{-i\kappa_i x_1} + \tilde{B}(f)e^{i\kappa_r x_1} \\ \tilde{P}(x_2, f) = \tilde{A}(f)e^{-i\kappa_i x_2} + \tilde{B}(f)e^{i\kappa_r x_2} \end{cases} \quad (2.68)$$

This linear system of equations can be solved for \tilde{A} and \tilde{B} . The reflection coefficient can then be computed by recalling its definition

$$R(x) = \frac{\tilde{P}_r(x, f)}{\tilde{P}_i(x, f)} = \frac{\tilde{B}(f)e^{i\kappa_r x}}{\tilde{A}(f)e^{-i\kappa_i x}} \quad (2.69)$$

which, given our convenient choice of the coordinate system, reduces to

$$R = \frac{\tilde{B}(f)}{\tilde{A}(f)} \quad (2.70)$$

Notice that this is the classical definition of the reflection coefficient, which differs to the one presented in Seybert and Ross [29], where they refer to the power reflection coefficient. The way in which the MMM works is exactly the same as the TMM described so far, but instead of solving two equations for two unknowns, we solve as many equations as the number of microphones. This implies solving an undetermined system of equations similar to Eq. (2.68), where now \tilde{A} and \tilde{B} are obtained as best-fit values of a regression problem.

The numerical implementation of the base flow model is reported in a Matlab[®] routine that can be found in Appendix D. Note that the flow velocity in our experiment is $u \sim \mathcal{O}(10^{-1})$, as compared with the propagation speed $c_0 \sim \mathcal{O}(10^2)$. Hence, the incident and reflected wavenumbers can be safely computed as $\kappa_i = \kappa_r = \omega/c_0$. An important consideration is that introducing the heater changes the acoustics of the tube, because heat release fluctuations interact with acoustic fluctuations and because any drag at the heater causes a pressure jump across the heater. This means that, in order for the approach illustrated so far to be valid, the microphones need to be placed only downstream of the heater. Because the heater location in this experiment is $x/L = 0.25$, this requirement is fulfilled only by Mic. 1 to 5, as shown by Table 2.1, therefore one should use only the signals coming from these microphones for the computation of the reflection coefficient. Section 3.3 shows what happens when all the microphones are taken into account. Another point is that the speed of propagation with zero flow is computed using the ideal gas law as $c_0 = \sqrt{\gamma \frac{R}{M_m} \bar{T}}$, where R is the universal gas constant, M_m is the air molar mass, and \bar{T} is the average temperature downstream of the heater when using the first five microphones, and in the entire tube when using all the six microphones. As a concluding remark, one should bear in mind that the reflection coefficient results from (i) a theoretical model that relies on simplifying assumptions (e.g. 1D acoustics, nondispersive system, etc.); (ii) a numerical computation of \bar{T} that is affected by numerical errors; and (iii) on experimental data through pressure measurements, which are inevitably affected by experimental errors. All these errors and simplifying assumptions result in some uncertainty, whose importance will be discussed in Chapter 3.

2.5 Thermoacoustic model

This section is devoted to deriving the governing equations of the acoustics inside the duct, and describing the adopted thermoacoustic model, highlighting the connection with the previous sections of this chapter. The idea is to use a wave-based low order network model, following the approach proposed by Stow and Dowling [10], and Aguilar et al. [11]. In our case, we consider a one dimensional network model that consists of a duct of length L with a compact heat source located at $x = b$. Homogeneous properties are assumed along each segment, so that the domain is split into two segments, as shown in Fig. 2.5. Each segment is governed by the same set of equations, which are connected by the jump conditions imposed by the heat source. Notice that assuming homogeneous properties is not realistic, as the gas temperature profile inside the duct varies significantly with the axial coordinate (e.g. from 300 K to 700 K). However, we make this assumption

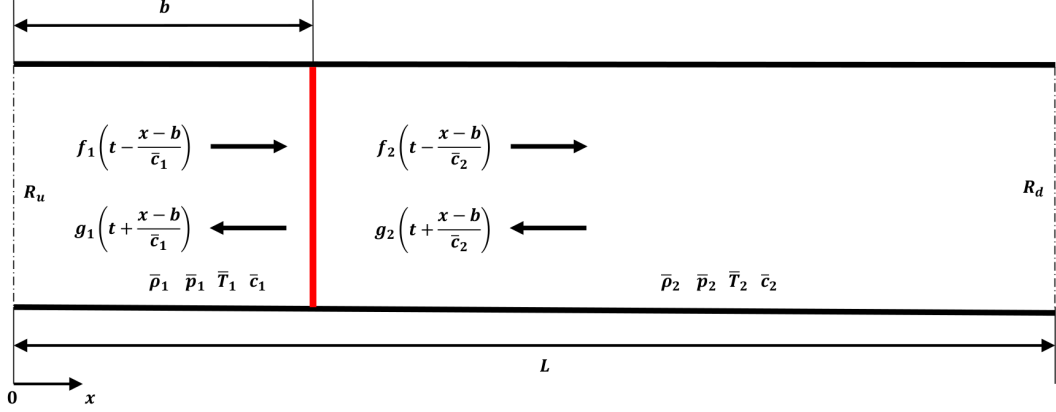


Figure 2.5: Network model of a tube of length L with a heat source located at $x = b$. The arrows represent waves travelling at the local speed of propagation. These waves reflect off the upstream and downstream ends, characterised by reflection coefficients R_u and R_d , respectively.

anyway, because it considerably simplifies the problem.

The Navier-Stokes equations, in the assumption of negligible viscosity and heat conduction, reduce to the Euler equations, which, upon neglecting gravity and considering one dimensional flow, read

$$\frac{\partial \rho}{\partial t} + u \frac{\partial \rho}{\partial x} + \rho \frac{\partial u}{\partial x} = 0 \quad (2.71)$$

$$\rho \frac{\partial u}{\partial t} + \rho u \frac{\partial u}{\partial x} + \frac{\partial p}{\partial x} = 0 \quad (2.72)$$

$$\frac{\partial p}{\partial t} + u \frac{\partial p}{\partial x} + \gamma p \frac{\partial u}{\partial x} = q(\gamma - 1)\delta(x - b) \quad (2.73)$$

where q is the heat release rate per unit area, and $\delta(x - b)$ is the Dirac delta function centred at $x = b$. Integrating these equations across the heat source, i.e. from $x = b^-$ to $x = b^+$, and assuming no accumulation of mass, momentum and energy, lead to the following set of jump conditions

$$[\rho u]_{b^-}^{b^+} = 0 \quad (2.74)$$

$$\left[p + \rho u^2 \right]_{b^-}^{b^+} = 0 \quad (2.75)$$

$$\left[\frac{\gamma}{\gamma - 1} p u + \frac{1}{2} \rho u^3 \right]_{b^-}^{b^+} = q(b, t) \quad (2.76)$$

We now write each of the three primitive variables (ρ , u , p) as a sum of a mean flow quantity, which depends neither on time nor on space due to the homogeneity

assumption, and a quantity fluctuating in time and space (i.e. $p(x, t) = \bar{p} + p'(x, t)$, etc.). Assuming planar acoustic perturbations prevents entropy and vorticity waves from propagating [31]. As discussed in Section 2.4.2, although the mean flow is non-zero, we can safely assume the Mach number to be zero (i.e. $\bar{u} = 0$) due to the different orders of magnitude between the mean flow velocity and the speed of propagation. Neglecting second-order terms allows us to derive the acoustics equation for mass, momentum, and energy, which, in the assumption of zero Mach number, read as follows

$$\frac{\partial \rho'}{\partial t} + \bar{\rho} \frac{\partial u'}{\partial x} = 0 \quad (2.77)$$

$$\bar{\rho} \frac{\partial u'}{\partial t} + \frac{\partial p'}{\partial x} = 0 \quad (2.78)$$

$$\frac{\partial p'}{\partial t} + \gamma \bar{p} \frac{\partial u'}{\partial x} = q'(\gamma - 1)\delta(x - b) \quad (2.79)$$

Linerization of the governing equations, together with the $\bar{u} = 0$ assumption, leads us to conclude that $[\bar{p}]_{b^-}^{b^+} = 0$ (see Eq. (2.75)), i.e. the pressure is continuous everywhere in the duct. This follows from the assumption of neglecting drag at the wire location. In addition to this, one should also notice that the form of Eq. (2.77) resembles that of Eq. (2.79). In fact, by substituting $\rho' = p'/c^2$ into Eq. (2.77) we get exactly Eq. (2.79), provided that one recalls the ideal gas law result $\gamma p = \bar{p} \bar{c}^2$. This means that there are only two independent equations, Eq. (2.78) and Eq. (2.79), which we have to solve together with the following jump conditions

$$[p']_{b^-}^{b^+} = 0 \quad (2.80)$$

$$[u']_{b^-}^{b^+} = \frac{\gamma - 1}{\gamma} \int_{b^-}^{b^+} q' dx \quad (2.81)$$

Solution to Eqs. (2.78) to (2.81) requires a thermoacoustic relationship between q' and (u', p') to be known. In the present context, we use an $n - \tau$ model [12, 5], according to which heat release fluctuations are assumed to be proportional to velocity fluctuations through an interaction index n , and time-delayed with respect to them by a factor τ , as expressed by the following equation

$$q'(x, t) = nu'(b^-, t - \tau)\delta(x - b) \quad (2.82)$$

We can re-write Eqs. (2.78) and (2.79) in matrix form as $\mathbf{u}_t + A\mathbf{u}_x = \mathbf{0}$, i.e.

$$\begin{bmatrix} u' \\ p' \end{bmatrix}_t + \begin{bmatrix} 0 & 1/\bar{\rho} \\ \gamma\bar{p} & 0 \end{bmatrix} \begin{bmatrix} u' \\ p' \end{bmatrix}_x = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (2.83)$$

This is a hyperbolic system of equations, which can be solved by making a change of variables that allows us to decouple the two equations. To do so, we proceed by first diagonalising matrix A as follows

$$A = S\Lambda S^{-1} = \begin{bmatrix} 1/(\bar{\rho}\bar{c}) & -1/(\bar{\rho}\bar{c}) \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \bar{c} & 0 \\ 0 & -\bar{c} \end{bmatrix} \begin{bmatrix} \bar{\rho}\bar{c}/2 & 1/2 \\ -\bar{\rho}\bar{c}/2 & 1/2 \end{bmatrix} \quad (2.84)$$

We then substitute Eq. (2.84) into Eq. (2.83) and pre-multiply each term by S^{-1} , thus obtaining

$$S^{-1}\mathbf{u}_t + \Lambda S^{-1}\mathbf{u}_x = \mathbf{0} \quad (2.85)$$

The two equations can now be decoupled upon using $\mathbf{v} = S^{-1}\mathbf{u}$, so that Eq. (2.85) becomes $\mathbf{v}_t + \Lambda\mathbf{v}_x = \mathbf{0}$, which reads

$$\begin{bmatrix} \mathcal{U} \\ \mathcal{P} \end{bmatrix}_t + \begin{bmatrix} \bar{c} & 0 \\ 0 & -\bar{c} \end{bmatrix} \begin{bmatrix} \mathcal{U} \\ \mathcal{P} \end{bmatrix}_x = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (2.86)$$

Each of these two equations is a simple wave equation, and the vector $\mathbf{v} = [\mathcal{U}, \mathcal{P}]^T$ represents the Riemann invariants of the system, i.e. the travelling waves. The solution to the system is given by

$$\mathcal{U}(x, t) = f\left(t - \frac{x}{\bar{c}}\right) \quad \text{forward travelling wave} \quad (2.87)$$

$$\mathcal{P}(x, t) = g\left(t + \frac{x}{\bar{c}}\right) \quad \text{backward travelling wave} \quad (2.88)$$

The solution to the original system, Eq. (2.83), can be obtained by re-transforming the variables according to $\mathbf{u} = S\mathbf{v}$. By considering that the domain is split into two sub-domains, each one denoted by subscripts 1 or 2, and that the split occurs at the heater location, $x = b$, pressure and velocity fluctuations become

$$p'(x, t) = \begin{cases} f_1\left(t - \frac{x-b}{\bar{c}_1}\right) + g_1\left(t + \frac{x-b}{\bar{c}_1}\right) & \text{if } x < b \\ f_2\left(t - \frac{x-b}{\bar{c}_2}\right) + g_2\left(t + \frac{x-b}{\bar{c}_2}\right) & \text{if } x > b \end{cases} \quad (2.89)$$

$$u'(x, t) = \begin{cases} \frac{1}{\bar{\rho}_1\bar{c}_1} \left[f_1\left(t - \frac{x-b}{\bar{c}_1}\right) - g_1\left(t + \frac{x-b}{\bar{c}_1}\right) \right] & \text{if } x < b \\ \frac{1}{\bar{\rho}_2\bar{c}_2} \left[f_2\left(t - \frac{x-b}{\bar{c}_2}\right) - g_2\left(t + \frac{x-b}{\bar{c}_2}\right) \right] & \text{if } x > b \end{cases} \quad (2.90)$$

We have obtained an expression for pressure and velocity fluctuations inside the duct in terms of forward and backward travelling waves, f and g . However, our real interest is, given a certain operating condition and boundary conditions, whether an infinitely small perturbation to the system grows or decays in time. If it grows the system is said to be (linearly) unstable, whereas if it decays the system is referred to as (linearly) stable. To address this question, it is convenient to Laplace-transform the unsteady variables, i.e. $f_1 = F_1e^{st}$ and $g_1 = G_1e^{st}$ (and similarly for f_2 and g_2).

We then match the values of F_1 and G_1 , and F_2 and G_2 at $x = b$, after reflection off the boundaries

$$F_1(s) = R_u(s)G_1(s)e^{-s\tau_u} \quad \tau_u = 2b/\bar{c}_1 \quad (2.91)$$

$$G_2(s) = R_d(s)F_2(s)e^{-s\tau_d} \quad \tau_d = 2(L - b)/\bar{c}_2 \quad (2.92)$$

Thus, the linearised jump conditions expressed in Eqs. (2.80) and (2.81) can be expressed as

$$\begin{bmatrix} -1 - R_u e^{-s\tau_u} & 1 + R_d e^{-s\tau_d} \\ \frac{1}{\bar{\rho}_1 \bar{c}_1} (1 - R_u e^{-s\tau_u}) \left(1 + \frac{\gamma-1}{\gamma \bar{p}_1} n e^{-s\tau}\right) & \frac{1}{\bar{\rho}_2 \bar{c}_2} (1 - R_d e^{-s\tau_d}) \end{bmatrix} \begin{bmatrix} G_1 \\ F_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (2.93)$$

Notice that, because we have assumed no pressure losses we have that $\gamma \bar{p}_1 = \bar{\rho}_1 \bar{c}_1^2 = \bar{\rho}_2 \bar{c}_2^2 = \gamma \bar{p}_2$, which implies that Eq. (2.93) can also be written as

$$\begin{bmatrix} -1 - R_u e^{-s\tau_u} & 1 + R_d e^{-s\tau_d} \\ (1 - R_u e^{-s\tau_u}) \left(1 + \frac{\gamma-1}{\gamma \bar{p}_1} n e^{-s\tau}\right) & \frac{\bar{c}_2}{\bar{c}_1} (1 - R_d e^{-s\tau_d}) \end{bmatrix} \begin{bmatrix} G_1 \\ F_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (2.94)$$

which is the form reported in Aguilar et al. [11].

The system of equations just derived can be written as $L(s)\hat{\mathbf{q}} = \mathbf{0}$ and represents a nonlinear eigenvalue problem in s . Non-trivial solutions to Eq. (2.93) are obtained by setting the determinant of L to 0. This allows us to calculate the eigenvalues s_j , through which the eigenvectors $\hat{\mathbf{q}}_j$, and subsequently the mode shapes (\hat{p}, \hat{u}) , can be computed. Note that the eigenvalues s_j are complex numbers, whose real part physically represents the growth (> 0) or decay (< 0) rate of infinitely small perturbations, and whose imaginary part represents their frequency.

The goal of this project is to find a low-order model that, when coupled with data assimilation, becomes quantitatively accurate at predicting the system instability over a wide range of operating conditions. Treating the model parameters n and τ as constants over the entire range represents a too simplistic approach. A better approach is to assume n and τ to depend on problem variables such as input power, reference velocities, etc. We expect that the larger the input power to the system, the stronger the coupling between heat release fluctuations and velocity fluctuations. The easiest way to express this relationship is by assuming a linear dependency of the interaction index n on the input power q . The time delay τ is instead assumed to be proportional to $0.2d_w/U_1$, where d_w is the wire diameter of the heater, whereas U_1 is the inlet bulk velocity. This relationship is suggested by Lighthill [32], who performed a perturbation analysis on the boundary layer of a laminar flow around a cylinder, and found $\tau = 0.2d_w/U_1$. A proportionality constant is included to account for modeling errors, uncertainty related to the measurement of d_w , and the fact that the heater consists of many wires. A summary of the model is provided by the following equations

$$n = k_n q \quad (2.95)$$

$$\tau = k_\tau \frac{0.2d_w}{U_1} \quad (2.96)$$

The base flow model enters the thermoacoustic problem through the values of \bar{c}_1 , \bar{c}_2 and U_1 . The reflection coefficients R_u and R_d are assumed to be equal, and their value is computed using the multi-microphone method, which, in turn, relies on the base flow model through the value of \bar{c}_2 (or \bar{c}), as explained in Section 2.4.2. As stated in Section 2.4.1, the decay rate of the oscillations, together with their frequency, is experimentally determined. This means that 1080 experimentally observed eigenvalues, s_{exp} , are available. The idea is therefore to find the values of k_n and k_τ that best explain the experimental observations. This is done through simple regression, and details will be provided in Section 2.6. If the model is good enough, the eigenvalues s should compare well with the experimentally determined eigenvalues s_{exp} .

Although regression is done, it is not guaranteed to find good agreement between model predictions and experiments if the model is not sufficiently good. Once a good fitting is achieved, the idea is to use the *learned* parameters of both the base flow model and the thermoacoustic model to extrapolate results at different *testing* conditions, in which the system configuration is changed, e.g. by changing the heater location or the outlet diameter by means of a controllable iris. (This part is not addressed in this study - it will be done at a later stage.) Note that *learning* and *testing* are two words commonly used by the Machine Learning community. However, this approach is not data-driven based as intended in proper Machine Learning - it is rather physics-based Machine Learning, where the hidden relationship between inputs and outputs relies on physical arguments rather than on hidden layers. The advantage of this approach is that, in principle, it allows for more flexibility, in the sense that extrapolation should be safer, as the most influential physics is involved. The drawbacks are that it requires a good physical knowledge of the system in order to be successful, and that a low-order model is not guaranteed to be able to properly describe the system dynamics.

2.6 Problem setting

The last part that needs to be covered from a theoretical point of view is determination of the best-fit k_n and k_τ . In Section 2.5 it was stated that 1080 experimentally observed eigenvalues, s_{exp} , are available. Reflection coefficients R_u and R_d , inlet gas velocity U_1 , and average speed of propagation in the two sections of the domain \bar{c}_1 and \bar{c}_2 are also available. This means that we have a set of 1080 eigenvalue problems given by Eq. (2.94), where the unknowns are k_n , k_τ , and s . Ideally one would like to find a pair of k_n and k_τ such that for each eigenvalue problem $s = s_{exp}$ and, at the same time, the determinant of the eigenmatrix is zero, i.e. $|L|=0$. This would happen only if the model was able to completely explain the physics of the system, which is never the case, and at the same time if the experiments were error-free, which in practice never occurs either. Therefore, one has to accept a certain level of discrepancy in at least one of the two requirements

mentioned above. Notice that the requirement $|L|=0$ translates into the following equation

$$\begin{aligned} & \left[-1 - R_u e^{-s\tau_u}\right] \left[\frac{\bar{c}_2}{\bar{c}_1} (1 - R_d e^{-s\tau_d})\right] \\ & - \left[(1 - R_u e^{-s\tau_u}) \left(1 + \frac{\gamma - 1}{\gamma \bar{p}_1} n e^{-s\tau}\right) \right] \left[1 + R_d e^{-s\tau_d}\right] = 0 \end{aligned} \quad (2.97)$$

Based on these considerations, one could elaborate a first approach that works as follows. Starting from a first-guess pair of values (k_n, k_τ) one could solve Eq. (2.97) for s with a method that finds the zeros of a non-linear function, such as Newton's method. A good first-guess value for s , needed for the Newton's method, is given by experiments, as s_{exp} is known. Newton's method is quite fast and converges in 4-5 iterations. However, for every pair of parameters (k_n, k_τ) , Newton's method's routine shall be called 1080 times to solve as many different eigenproblems. The algorithm explores different pairs of parameters until the sum of squared difference between s and s_{exp} is minimised. For such purpose, an optimisation algorithm such as the Levenberg-Marquardt algorithm may be used. The idea in which the best-fit (k_n, k_τ) are found according to this approach is summarised by the following expression

$$\min_{k_n, k_\tau} \sum_{j=1}^{1080} (s_j - s_{exp,j})^2$$

Notice that, by means of this approach, the condition $|L|_j=0$ is satisfied for any j (within numerical tolerance), but in general $s_j \neq s_{exp,j}$. It is worth stressing once more that, although the Newton's method for the single eigenvalue problem is quite fast, the overall approach is very inefficient, especially if the algorithm is started from first-guess values that are far from the optimal values.

A more efficient approach is suggested by the need of speeding up the optimisation routine. The idea is to avoid using Newton's method to compute s . In this approach, each eigenvalue s_j is set to be equal to the corresponding experimental eigenvalue, $s_{exp,j}$, and (k_n, k_τ) are obtained from the following expression

$$\min_{k_n, k_\tau} \sum_{j=1}^{1080} |L_j(k_n, k_\tau)|^2$$

As opposed to the first approach, we now have that $s_j = s_{exp,j}$ for any j , but in general $|L_j| \neq 0$. The optimisation algorithm that may be used in this case is again the Levenberg-Marquardt algorithm, which is already available as inbuilt Matlab[®] function. To conclude, both approaches are sensible, but the second one is much more efficient than the first one (it only takes a few seconds). Therefore this is the approach that is used to produce the results in this context.

We now discuss details regarding the numerical implementation of the optimisation routine. First of all, we re-write Eq. (2.97) as follows

$$\frac{(\gamma - 1)}{\gamma \bar{p}_1} n e^{-s\tau} = \left[-1 - \frac{\bar{c}_2 (1 - R_d e^{-s\tau_d} + R_u e^{-s\tau_u} - R_u R_d e^{-s(\tau_u + \tau_d)})}{\bar{c}_1 (1 - R_u e^{-s\tau_u} + R_d e^{-s\tau_d} - R_u R_d e^{-s(\tau_u + \tau_d)})} \right] \quad (2.98)$$

and then we define

$$RHS \triangleq \left[-1 - \frac{\bar{c}_2 (1 - R_d e^{-s\tau_d} + R_u e^{-s\tau_u} - R_u R_d e^{-s(\tau_u + \tau_d)})}{\bar{c}_1 (1 - R_u e^{-s\tau_u} + R_d e^{-s\tau_d} - R_u R_d e^{-s(\tau_u + \tau_d)})} \right] \quad (2.99)$$

The values of k_n and k_τ we seek for are real and non-negative, for physical reasons. However, Eq. (2.98) contains complex variables (s , R_u , R_d). This means that, in order to obtain real values of the two parameters, it is necessary to split this equation into two different equations, one for the real and for the imaginary part, and solve them simultaneously. This leads to the following system of real equations

$$\begin{cases} RHS_r - \frac{(\gamma-1)}{\gamma p_1} n e^{-\tau s_r} \cos(\tau s_i) = 0 \\ RHS_i + \frac{(\gamma-1)}{\gamma p_1} n e^{-\tau s_r} \sin(\tau s_i) = 0 \end{cases} \quad (2.100)$$

where the subscripts r and i stand for real and imaginary part, respectively. In the numerical implementation, the right-hand sides of the two equations that make up this system are set to be equal to a residual value, ε_1 and ε_2 , respectively, and k_n and k_τ are found as follows

$$\min_{k_n, k_\tau} \sum_{j=1}^{1080} (\varepsilon_{1,j}^2 + \varepsilon_{2,j}^2)$$

The function that computes the residual values is

```
function res_output = f_obj(params)

    dw = 0.559e-3; % wire diameter [m]
    p1 = 101325; % ambient pressure [Pa]
    gamma = 1.4; % specific heat capacity ratio for air

    load('regressionVariables.mat');
    load('RHS.mat');

    n = params(1)*q;
    tau = params(2)*0.2*dw./u1;

    eps1 = RHS_r - (gamma - 1)/gamma/p1*n.* ...
        exp(-tau.*real(s)).*cos(tau.*imag(s));

    eps2 = RHS_i + (gamma - 1)/gamma/p1*n.* ...
        exp(-tau.*real(s)).*sin(tau.*imag(s));

    res_output = [eps1 eps2];

end
```

The above function is then passed to the optimisation routine, which is implemented as follows

```

opt = optimoptions('lsqnonlin','Algorithm','levenberg-marquardt');
params0 = [100,1]; % First-guess values of kn and ktau
[params,resnorm] = lsqnonlin(@res_output,params0,[],[],opt)

```

Once the best-fit k_n and k_τ are computed, one can use Newton's method to obtain a-posteriori s_j for each eigenvalue problem, and compare them to $s_{exp,j}$. Newton's method implementation is reported in Appendix E.

2.7 Summary

This short section summarises the main steps in order to show how the different parts of the project are connected.

- i) Carry out an automated experiment by changing the input power by 10 W every 68 minutes, spanning the range 10 W - 180 W. During the experiment, record every 8 seconds gas and solid temperatures at different locations. Moreover, every 68 seconds, pulse the system at a frequency close to the natural frequency of the tube, and observe the oscillations' decay. With the aid of six microphones record this decay.
- ii) By guessing an initial set of parameters (Nu_i , Nu_o and k_i), run the base flow model code so as to reproduce the experiment dynamics, and, by means of the Ensemble Kalman Filter, assimilate every 8 seconds temperature data measured at the previous step. We now have an estimate of state (gas and solid temperatures, and inlet gas velocity) and parameters (Nu_i , Nu_o and k_i) every 8 seconds, together with their uncertainty.
- iii) Compute the average speed of propagation in the upstream and downstream sections of the tube, namely \bar{c}_1 and \bar{c}_2 , every 68 seconds. Use the value of \bar{c}_2 if using only the microphones downstream of the heater, or \bar{c} if using all six microphones, as input to the multi-microphone method routine, which, together with the pressure measurements obtained at step i), is used to compute the reflection coefficient at the top end of the tube, R_d , every 68 seconds. Assume R_d to be equal to the reflection coefficient at the bottom end of the tube, R_u .
- iv) Extract from the pressure measurements of step i) decay rate and frequency of the oscillations just after the pulse. These represent the real and imaginary parts, respectively, of the experimental eigenvalues, s_{exp} . Because the system is pulsed every 68 seconds, a total of 1080 experimental observations are now available for regression.
- v) Use the reflection coefficients, R_u and R_d , the average speed of propagation in the two sections of the tube, \bar{c}_1 and \bar{c}_2 , and the inlet gas velocity, U_1 , as input to the nonlinear eigenvalue problem given by Eq. (2.94). Find the values of k_n and k_τ that best explain the experimental observations, s_{exp} , using a nonlinear regression algorithm.

- vi) Use the best-fit k_n and k_r to compute a-posteriori the model predictions, in terms of the eigenvalue s at each operating condition. Compare these to the experimental eigenvalues, s_{exp} , and draw conclusions on the validity of the approach.

It is worth also mentioning how the project will proceed in the future, although results are not yet available for this part.

- vii) Once a good thermoacoustic model is found, and once the model parameters of both the base flow and the thermoacoustic models are known, extrapolate results to new configurations. In this case no data assimilation will be performed because the old parameters will be used. It is possible to change the system configuration by changing for example the heater location, or by changing the diameter of the outlet section through a controlled iris. The latter case requires some new experiments to infer the value of the reflection coefficient at the outlet section, as this is expected to strongly depend on the iris diameter.
- viii) If the model performs well on both the training and the testing set of operating conditions, couple it with adjoint-based sensitivity analysis to work out the small changes to apply to the system in order to stabilise it. These changes may again be related, for this simple configuration, to the outlet section diameter or to the heater location.
- ix) If all the previous steps give positive feedback, apply the same methodology to a more realistic system, (e.g. a Rijke tube with a flame instead of an electric heater, and later on a lab-scale combustor). The model parameters learned on the simpler system are used as inputs to study the more complicated systems.

Chapter 3

Results

3.1 Data assimilation

We present here results of the base flow model coupled with data assimilation. As stated in Section 2.3.1, two types of experiments have been carried out: in the first one the heater is inserted from the top, thus introducing additional acoustic damping, whereas in the second experiment the heater is inserted from the bottom (refer to Table 2.2 for details). Moreover, the first experiment has been carried out in summer, whereas the second experiment in autumn, when the ambient temperature was about 3 K lower. Fig. 3.1 shows the solid and gas temperatures along the axial location at a particular instant in time ($t = 14.2$ h from the start) for the two experiments. Notice that these profiles coincide with those obtained after the update step performed by the Ensemble Kalman Filter. Nevertheless, the uncertainty related to the model predictions is not reported in these plots, although this information is available. Results show a reasonable agreement if we consider that our simplified base flow model does not account for any radiation from the heater, which especially affects wall regions close to the heater location, as suggested by the experiments. Another source of error comes from the fact that the bulk temperature predicted by the 1D model in reality is not located at the tube centreline. Furthermore, the shape of the source term (see Fig. 2.1) is assumed Gaussian without having any knowledge on how it is in reality. All this represents source of discrepancy between model predictions and experimental observations. Notice that the second experiment has been set up in order to place more thermocouples on the gas side, so as to have more reliable estimates of the gas temperature profile. Indeed, what the thermoacoustic model requires is an estimate of the average gas temperature in the two sections of the tube, as well as a value for the inlet gas velocity.

Fig. 3.2 shows for both experiments the estimated base flow model parameters over time (and hence as the input power increases) with the related uncertainty, here plotted as an error bar of size $\pm 2\sigma$. Because the analysed state and parameters are Gaussian distributed, such an error bar corresponds to a probability of 95.5% of being in that range. The profiles are similar in the two cases. It is important to realise that, although one may have no knowledge of the values of such parameters, the filter is anyway able to find, after some iterations, the most likely estimate,

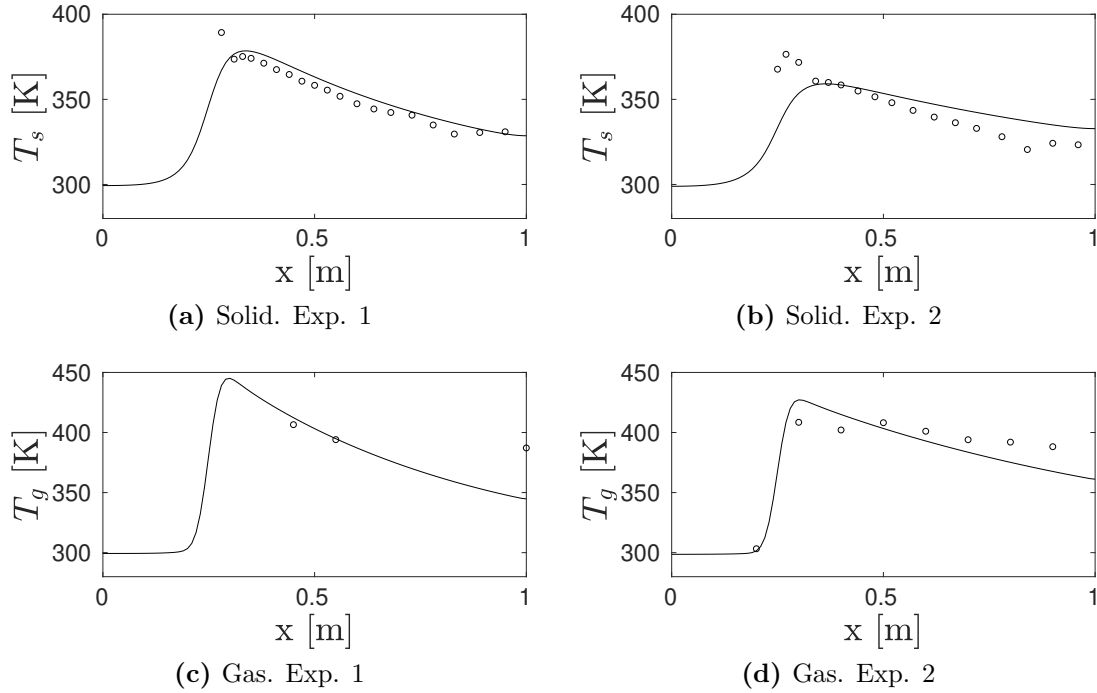
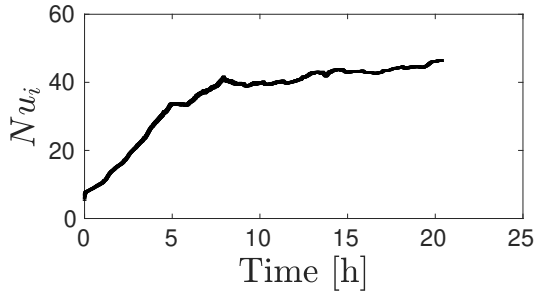


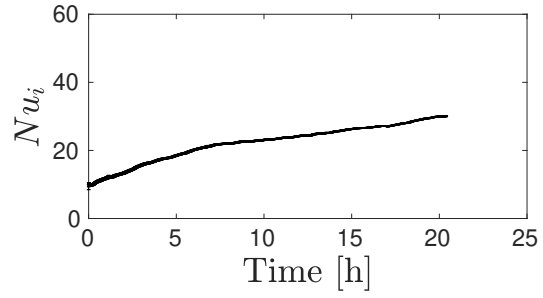
Figure 3.1: Solid (top) and gas (bottom) temperature profiles along the tube axis at $t = 14.2$ h (corresponding to $q = 120$ W). Exp. 1 (left): $T_a = 300.5$ K and heater from the top; Exp. 2 (right): $T_a = 297.5$ K and heater from the bottom. Circles are experiments, solid lines are model predictions

despite this significantly differing from the provided initial condition (see for example Figs. 3.2e and 3.2f, where κ_i was assumed to have an initial mean value of 55). Regarding the shape of these curves, if the model were exact, then the parameters would remain constant over time. Indeed, the pressure loss coefficient κ_i seems to be approximately constant. The same cannot be said for the two Nusselt numbers however, which increase as the input power increases. Nevertheless, this should not be considered a deficiency of the model, but rather a physical fact. Indeed, in natural convection the Nusselt number over a vertical flat plate is expected to grow with the buoyancy force [33], and the same occurs here, despite the geometry being different. Therefore, because the change in the Nusselt numbers can be justified by physical arguments, we proceed in our analysis. It is worth mentioning that, because the error bar size decreases as more data are assimilated into the model, especially for Nu_o and κ_i , it seems that convergence of the Ensemble Kalman Filter is achieved.

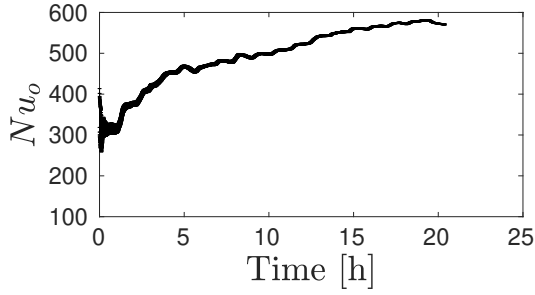
We conclude this section by showing in Fig. 3.3 the behaviour of the inlet gas velocity and the average sound speed in the two sections of the duct, upstream and downstream of the heater. These are the only three quantities of the base flow model that feed into the thermoacoustic model, but they need to be sufficiently accurate in order for the system's stability to be evaluated accurately. The slight increase of the sound for the first experiment compared to the second is due to the fact that, as highlighted above, the first experiment was conducted in summer, when the ambient temperature was about 3 K higher than that of the second



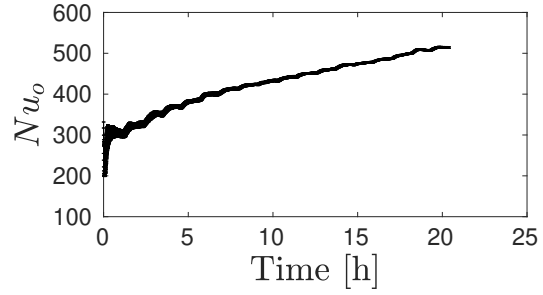
(a) Inner Nusselt number. Exp. 1



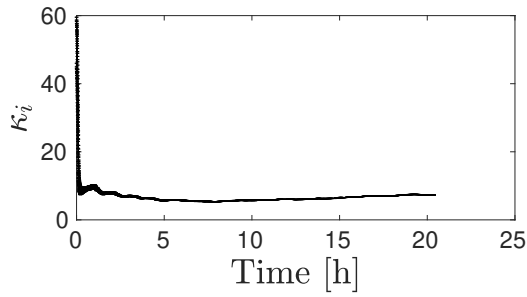
(b) Inner Nusselt number. Exp. 2



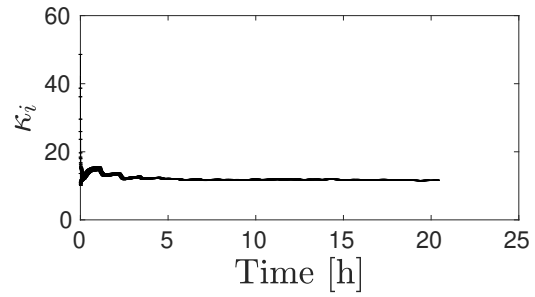
(c) Outer Nusselt number. Exp. 1



(d) Outer Nusselt number. Exp. 2



(e) Pressure loss coefficient. Exp. 1



(f) Pressure loss coefficient. Exp. 2

Figure 3.2: Error bar of size $\pm 2\sigma$ of the base flow model parameters. Exp. 1 (left): $T_a = 300.5$ K and heater from the top; Exp. 2 (right): $T_a = 297.5$ K and heater from the bottom

experiment. Physically, if the ambient temperature gets larger, the temperature difference ΔT between the inside of the tube and the surrounding ambient gets lower. This means that the buoyancy strength, which is proportional to the ratio $\Delta T/T_a$ and which drives the natural flow, is smaller, and therefore the inlet gas velocity is also smaller, as shown by Fig. 3.3a when compared to Fig. 3.3b.

Overall, although the two experiments are conducted in different periods of the year, and with a different number of thermocouples, they provide similar results for the base flow. This is important because it suggests that discrepancies between the stability evaluated in the two experiments is probably caused by the thermoacoustic model, rather than by the base flow model.

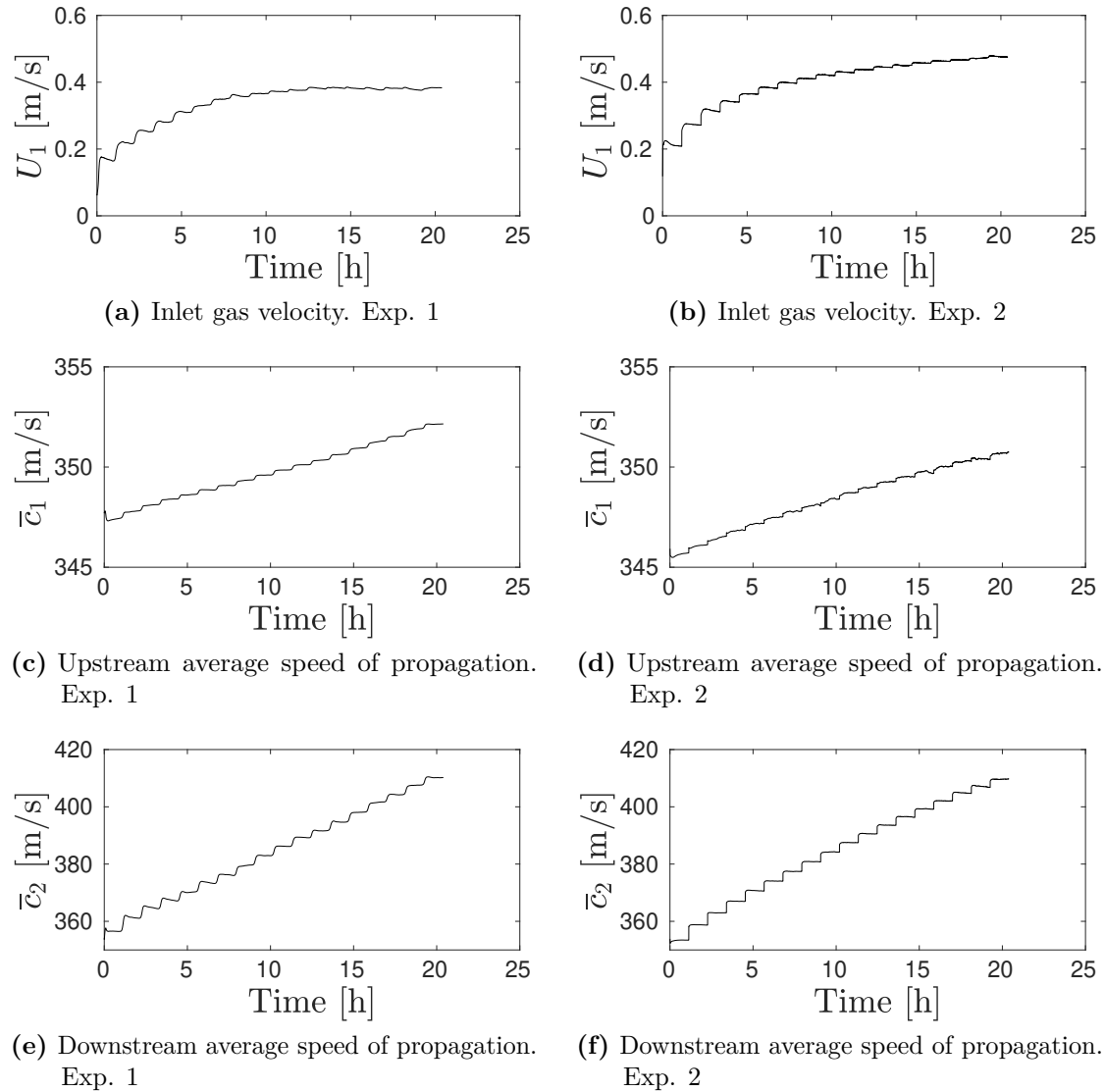


Figure 3.3: Inlet gas velocity and average speed of propagation upstream and downstream of the heater. Exp. 1 (left): $T_a = 300.5$ K and heater from the top; Exp. 2 (right): $T_a = 297.5$ K and heater from the bottom

3.2 Experimental eigenvalues

The second step is to show the experimental eigenvalues, obtained by following the approach described in Section 2.4.1. Fig. 3.4 shows the experimental growth rate and frequency of the oscillations for the two cases. The most striking feature is undoubtedly the difference in the decay rate, which in the first experiment is almost twice as large as in the second one, because of the increased acoustic damping due to the longer rods the heater is attached to. The frequency does not seem to vary significantly between the two experiments. It is worth remarking that our thermoacoustic model, as described in Section 2.5, does not account for any acoustic damping, except via the reflection coefficients. It is challenging to account for a distributed source of damping in a network model. The results that we present in Section 3.3 do not account for any damping, which will be discussed in the next section and investigated in future work.

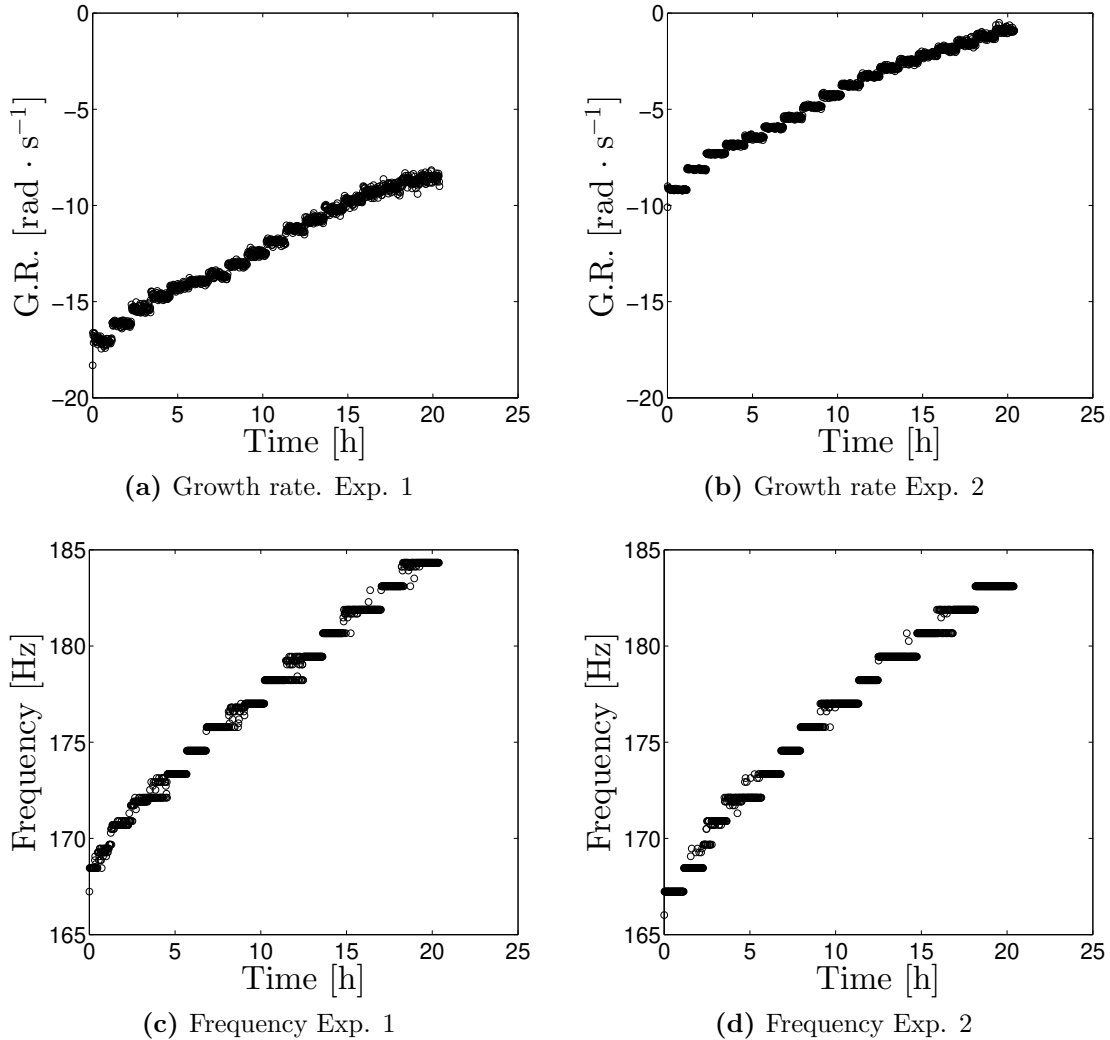


Figure 3.4: Experimental growth rate (top) and frequency (bottom) of the oscillations. Exp. 1 (left): $T_a = 300.5 \text{ K}$ and heater from the top; Exp. 2 (right): $T_a = 297.5 \text{ K}$ and heater from the bottom

3.3 Evaluation of the parameters of the acoustic model

For all the results, the values of k_n and k_τ obtained from regression are reported in the figure caption. We first compare in Fig. 3.5 the experimental observations with the model predictions based on the reflection coefficients obtained with the multi-microphone method using five microphones, all placed downstream of the heater, for the reasons illustrated in Section 2.4.2. The results for the first experiment are clearly wrong and the best fit k_n and k_τ differ greatly between the two experiments. Although one is tempted to conclude that the algorithm has converged to wrong parameters, the actual problem turns out to be the calculation of the reflection coefficient. We discussed in Section 2.4.2 that the multi-microphone method is affected by several types of errors, which all contribute to small variations in the experimental reflection coefficient. As proved by Table 3.1, these variations are indeed very small, but they have a huge impact on the results of the stability analysis. In particular, it is worth noting that the average reflection coefficient for the first experiment is equal to $R = -0.9701 + 0.1165i$. This will turn out to be useful later on.

	Exp. 1				Exp. 2			
	Re(R)	Im(R)	$ R $	$\angle R$	Re(R)	Im(R)	$ R $	$\angle R$
Min	-0.9745	0.1081	0.9684	3.0106	-0.9901	0.0910	0.9727	3.0200
Max	-0.9607	0.1274	0.9814	3.0310	-0.9660	0.1184	0.9956	3.0487
Mean	-0.9701	0.1165	0.9771	3.0221	-0.9727	0.1043	0.9783	3.0348

Table 3.1: Minimum, maximum, and mean values of real part, imaginary part, modulus, and angle (rad.) of the reflection coefficient R using Mic. 1, 2, 3, 4, 5

It is interesting to see what happens when six microphones are used for the determination of the reflection coefficient. The sixth microphone is placed at $x/L = 0.15$, upstream of the heater. As discussed in Section 2.4.2, it is not conceptually correct to compute the reflection coefficient accounting for such a microphone, because of the presence of the heater between the sixth microphone and the top end of the tube. Nonetheless, we do this anyway, in order to see whether there are significant changes in the results. Fig. 3.6 surprisingly shows very good agreement in the growth rate results of the first experiment, whereas results get worse for the second experiment. The frequency is correctly captured for both experiments. Table 3.2 reports details regarding the variation of the reflection coefficient in this case. The mean value of the reflection coefficient for the first experiment is now $R = -0.9473 + 0.1263i$, therefore similar to that highlighted above.

It is therefore legitimate to question whether such small variations in the reflection coefficient can really be the cause of this large variation in the results. We now show, for a simpler case, that the sensitivity to the reflection coefficient is indeed very high. In order to obtain an analytical result, we consider the limiting case of no heater inside the tube. In such a case, the nonlinear eigenvalue problem

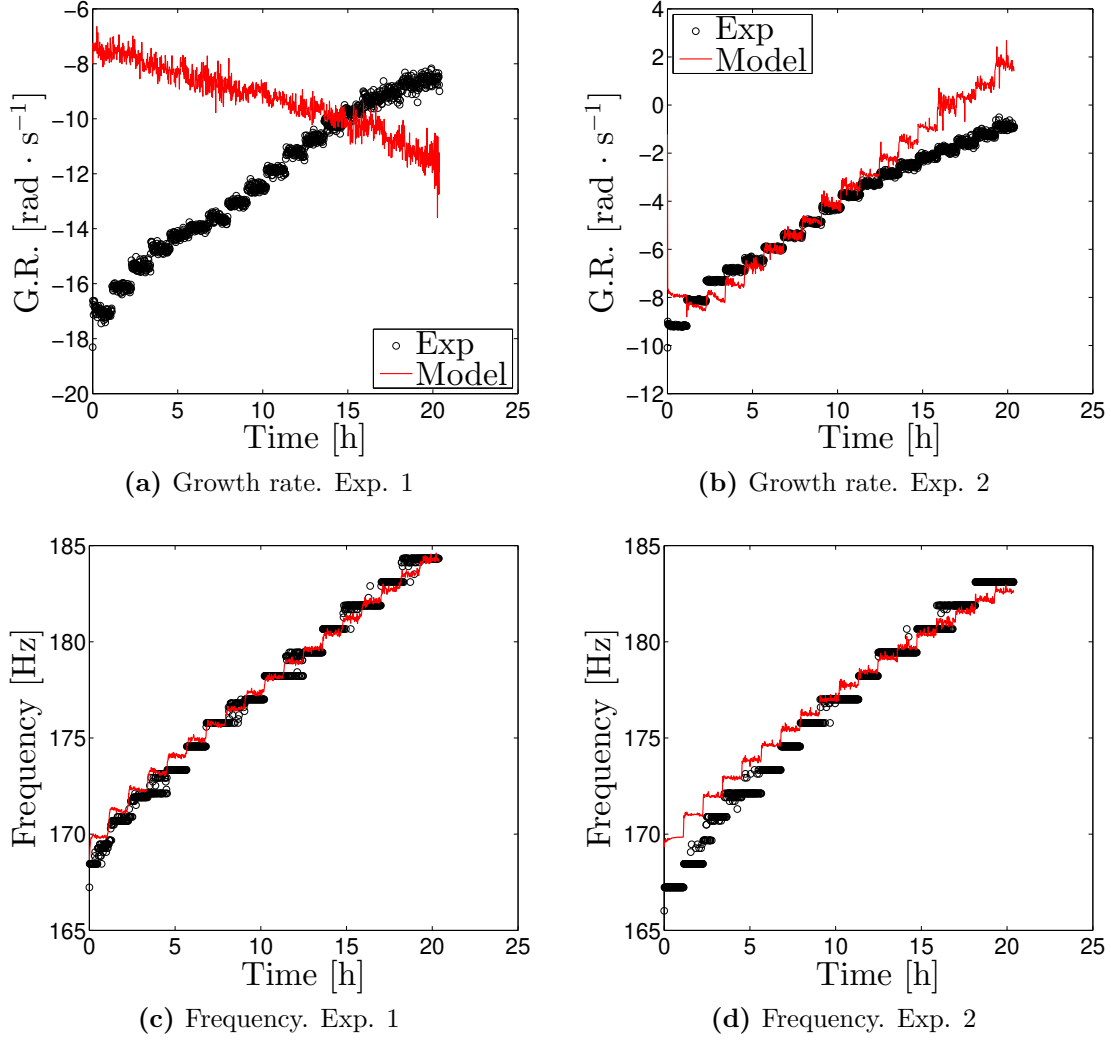


Figure 3.5: Experimental observations and model predictions of growth rate (top) and frequency (bottom) of the oscillations, using Mic. 1, 2, 3, 4, 5. Exp. 1 (left): $T_a = 300.5$ K and heater from the top; Exp. 2 (right): $T_a = 297.5$ K and heater from the bottom. Best-fit parameters for Exp. 1: $k_n = 21.01$ and $k_\tau = 10.84$. Best-fit parameters for Exp. 2: $k_n = 169.94$ and $k_\tau = 8.95$

	Exp. 1				Exp. 2			
	$\text{Re}(R)$	$\text{Im}(R)$	$ R $	$\angle R$	$\text{Re}(R)$	$\text{Im}(R)$	$ R $	$\angle R$
Min	-0.9548	0.1116	0.9491	2.9813	-0.9838	0.1101	0.9649	2.9982
Max	-0.9391	0.1520	0.9631	3.0246	-0.9562	0.1420	0.9940	3.0276
Mean	-0.9473	0.1263	0.9557	3.0090	-0.9601	0.1204	0.9677	3.0168

Table 3.2: Minimum, maximum, and mean values of real part, imaginary part, modulus, and angle (rad.) of the reflection coefficient R using Mic. 1, 2, 3, 4, 5, 6

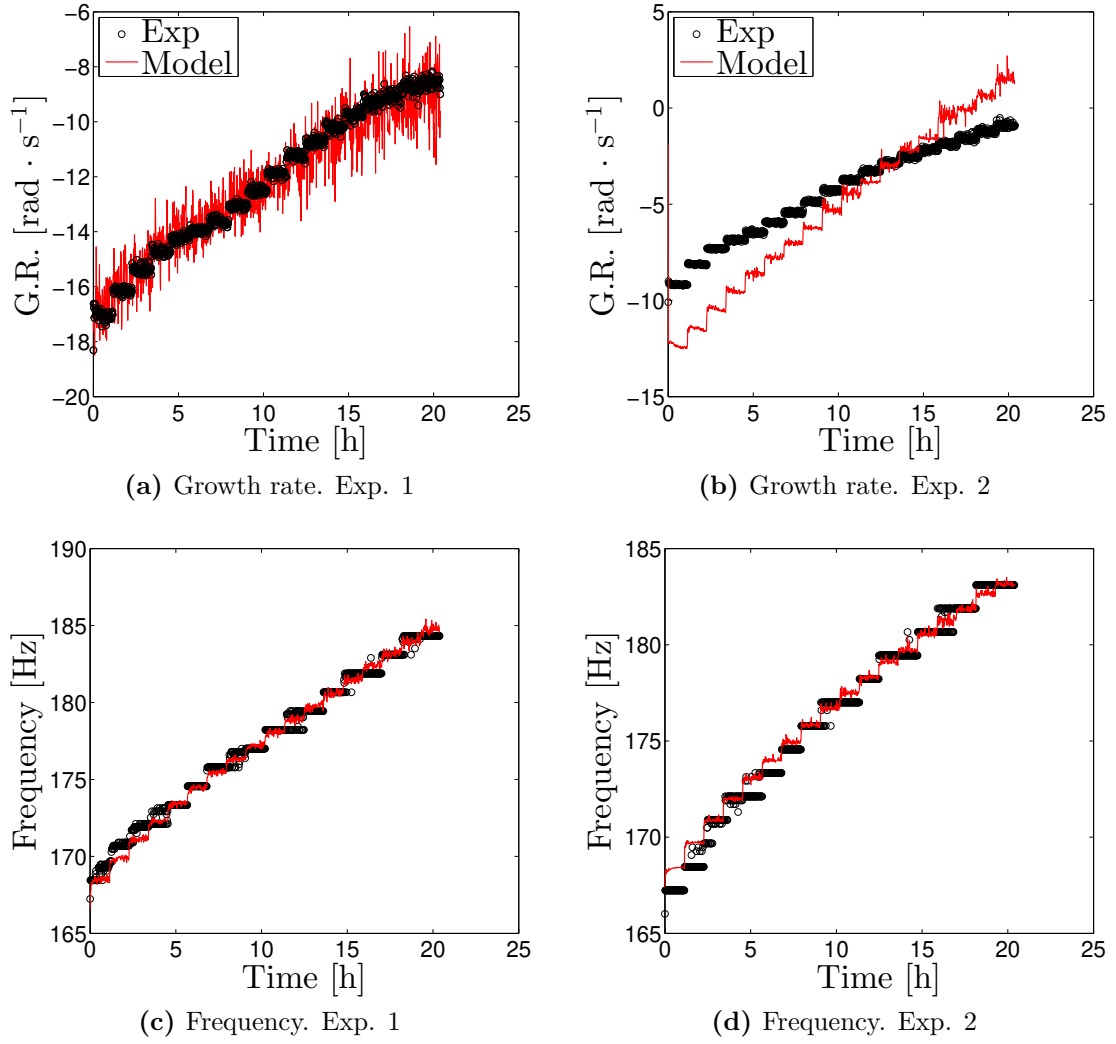


Figure 3.6: Experimental observations and model predictions of growth rate (top) and frequency (bottom) of the oscillations, using Mic. 1, 2, 3, 4, 5, 6. Exp. 1 (left): $T_a = 300.5$ K and heater from the top; Exp. 2 (right): $T_a = 297.5$ K and heater from the bottom. Best-fit parameters for Exp. 1: $k_n = 106.11$ and $k_\tau = 3.14$. Best-fit parameters for Exp. 2: $k_n = 155.55$ and $k_\tau = 6.67$

(see Eq. (2.94)) becomes

$$\begin{bmatrix} -1 - R_u e^{-s\tau_u} & 1 + R_d e^{-s\tau_d} \\ 1 - R_u e^{-s\tau_u} & 1 - R_d e^{-s\tau_d} \end{bmatrix} \begin{bmatrix} G_1 \\ F_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (3.1)$$

Setting the determinant of the above matrix to zero leads to the following expression

$$R_u R_d = e^{s(\tau_u + \tau_d)} \quad (3.2)$$

By recalling the definition of τ_u and τ_d (Eqs. (2.91) and (2.92)), and by considering that for the no-heating case $\bar{c}_1 = \bar{c}_2 = \bar{c}$, we can define a new timescale, τ_{tot} , such that $\frac{2L}{\bar{c}} = \tau_{tot} = \tau_u + \tau_d = \frac{2b}{\bar{c}_1} + \frac{2(L-b)}{\bar{c}_2}$. If we now assume $R_u = R_d$, as always done in this study, Eq. (3.2) becomes

$$R^2 = e^{s\tau_{tot}} \quad (3.3)$$

Implicit differentiation of the above equation leads to

$$2R dR = \tau_{tot} e^{s\tau_{tot}} ds = \tau_{tot} R^2 ds \quad (3.4)$$

which allows us to derive the following analytical expression for the derivative of the eigenvalue with respect to the reflection coefficient

$$\frac{ds}{dR} = \frac{2}{\tau_{tot} R} = \frac{\bar{c}}{RL} \quad (3.5)$$

This equation clearly shows that for the limiting case of no heater inside the tube, the eigenvalue sensitivity to small changes in the reflection coefficient is proportional to the average speed of sound inside the tube divided by the tube length, which in our case is larger than 340 s^{-1} . In the presence of a heater inside the tube, an equivalent expression is more complex to derive and does not provide straightforward insight, but it is reasonable to believe that the sensitivity is still high, given what observed so far.

To rigorously show that the problem lies in the reflection coefficient, we now analyse what happens when this coefficient is treated as a regression parameter, rather than being experimentally determined. By doing so, the regression parameters become k_n , k_τ , $\text{Re}(R)$, and $\text{Im}(R)$, being $\text{Re}(R)$ and $\text{Im}(R)$ the real and imaginary parts of R , respectively. As Fig. 3.7 shows, results are very good for both experiments. No fluctuations are present in the model predictions anymore, and there is good agreement between predictions and observations in experiments. The regression algorithm is very robust, in the sense that, even when starting from far initial guesses, it always converges to the same best-fit parameters. Notice that the best-fit thermoacoustic parameters found in both cases are not very different ($k_n = 106.30$, $k_\tau = 4.29$ vs. $k_n = 92.04$, $k_\tau = 5.95$), which is encouraging. Moreover, for the first experiment, the reflection coefficient that results from regression is $R = -0.9470 + 0.1191i$, which is very close to the average reflection coefficient obtained when using six instead of five microphones. (This is the reason why results shown in Fig. 3.6a are much better than those shown in Fig. 3.5a). In the

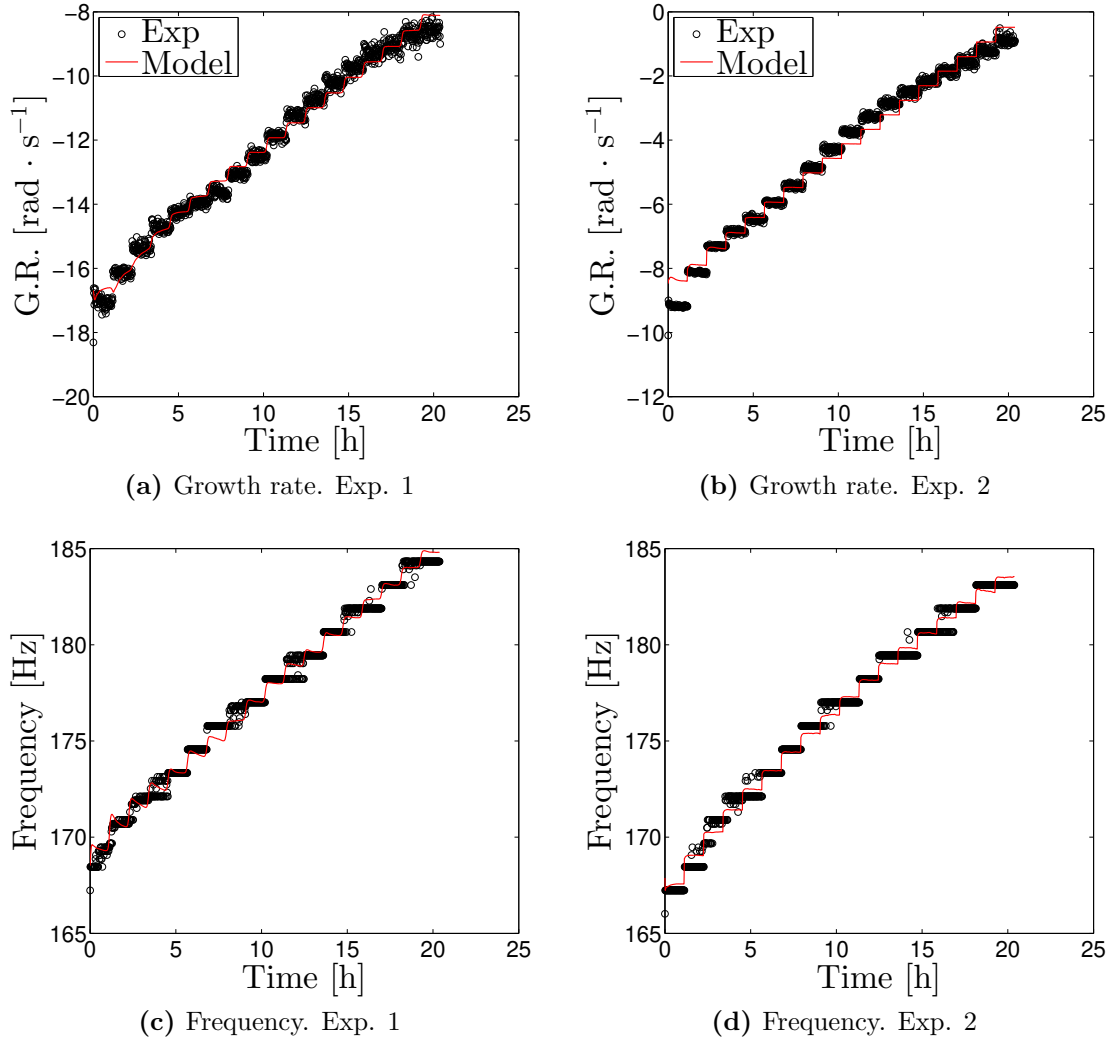


Figure 3.7: Experimental observations and model predictions of growth rate (top) and frequency (bottom) of the oscillations. Exp. 1 (left): $T_a = 300.5$ K and heater from the top; Exp. 2 (right): $T_a = 297.5$ K and heater from the bottom. Best-fit parameters for Exp. 1: $k_n = 106.30$, $k_\tau = 4.29$ and $R = -0.9470 + 0.1191i$. Best-fit parameters for Exp. 2: $k_n = 92.04$, $k_\tau = 5.95$ and $R = -0.9679 + 0.1305i$

comparison between Fig. 3.7a and Fig. 3.5a, the parameters that change are not only the reflection coefficient, but also k_n and k_τ . This means that it is still questionable whether the results improve mainly because the reflection coefficient changes, or because the thermoacoustic model parameters change. To answer this question, we provide results for what happens when the thermoacoustic model parameters are fixed to the best-fit values found in Fig. 3.7, and the reflection coefficient is equal to the one experimentally determined when using five microphones, as found in Fig. 3.5a. If the optimisation algorithm works well, then we should find a worse agreement with respect to what obtained in Fig. 3.5a, because for these values of the reflection coefficients, k_n and k_τ are not optimised. Indeed, we see in Fig. 3.8a that when optimisation is not performed, results get even worse. This demonstrates that the parameter that mostly affects the model predictions is indeed the reflection coefficient.

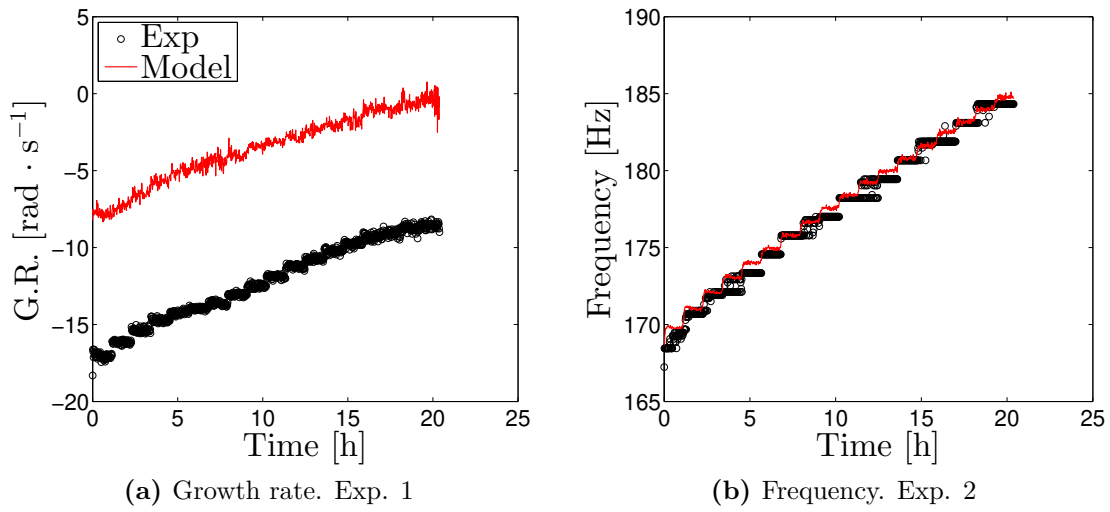


Figure 3.8: Experimental observations and model predictions of growth rate (left) and frequency (right) of the oscillations, using Mic. 1, 2, 3, 4, 5. Optimisation is not performed, and thermoacoustic parameters are fixed to: $k_n = 106.30$ and $k_\tau = 4.29$.

This conclusion highlights the most important deficiency of the present approach, that the multi-microphone method, in the way it is implemented in this context, is not suitable, because experimental observations will always be affected by intrinsic noise, which cannot be tolerated by our model. It is worth stressing that it is the model that is very sensitive to the boundary conditions, and not the physics of the problem. This is confirmed by the fact that by repeating the experiment (not shown here), we get very similar experimental eigenvalues, whereas the model predictions vary significantly because of small changes in the experimental reflection coefficient. More in general, our conclusion sheds light on the fact that it is not appropriate to characterise the reflection coefficient with one single value, because there is always uncertainty related to it. A better approach, to be tested in the future, would be to assign a probability density function to R , and propagate this uncertainty for the evaluation of the system stability. For each operating condition, this would result in an error bar associated to the growth rate and frequency predicted by the model,

rather than in one single value. Because the Ensemble Kalman Filter provides a mean value and associated uncertainty to state and parameters, an idea would be to use this tool for such a purpose. The state would be a two-dimensional vector, i.e. real and imaginary part of the eigenvalue, and the parameters would be k_n , k_τ , $\text{Re}(R)$, and $\text{Im}(R)$. Experimental eigenvalues could then be assimilated into the model every time they become available, exactly as done for the base flow model. The initial probability density functions for the filter could be obtained by taking the best-fit values obtained in Fig. 3.7 and providing a perturbation of 10% around these values, in a similar fashion to Avdonin et al. [14].

It is observed that when performing regression using the reflection coefficient as a parameter, then a very good agreement is found for both experiments, leading to similar thermoacoustic model parameters (k_n , k_τ). Because our model does not account for any acoustic damping, the fact that the agreement is good irrespective of the way in which the heater is inserted into the tube means that the effect of acoustic damping is accounted for by the change in the best-fit model parameters. This is of course a drawback of our approach, because ideally we would like to have fixed model parameters and at the same time be able to describe the physics observed for different configurations of the system. For this reason, in the future the possibility of including acoustic damping in the model will be explored. This may require us to add a new parameter to the existing thermoacoustic model, but it would allow us to have more flexibility in describing different system configurations. However, if we use a probabilistic approach as described above, we may not need to model the acoustic damping inside the tube at all, because then this effect will be implicitly absorbed by the probability density functions associated with the four existing parameters (k_n , k_τ , $\text{Re}(R)$, and $\text{Im}(R)$), similarly to that which already occurs for the two experiments in Fig. 3.7.

The approach used in the present study was already explored by Yu [34], who uses the same thermoacoustic model as the one adopted in this work, and treats the reflection coefficient as a regression parameter, as done in Fig. 3.7. However, a steady base flow model with heat losses is instead employed. Results obtained in [34] are shown in Fig. 3.9, and they should be compared to Figs. 3.7b and 3.7d. In addition to [34], this study includes (i) a revisited base flow model extended to the unsteady state, (ii) many more experiments performed at both steady and unsteady states, and (iii) the implementation of the Ensemble Kalman Filter for data assimilation in the base flow model. All these improvements have a positive effect on the results and prove that having an accurate base flow model, coupled with a technique for performing data assimilation is essential to get reliable estimates regarding the system stability. Moreover, the possibility of making predictions during transients is of great advantage because it saves time and provides many more data for training the model. In spite of the deficiencies still present in our approach, we conclude that this methodology seems to be working reasonably well and will certainly be improved in the near future when assimilating acoustic experimental data into the model by means of the Ensemble Kalman Filter, as proposed above.

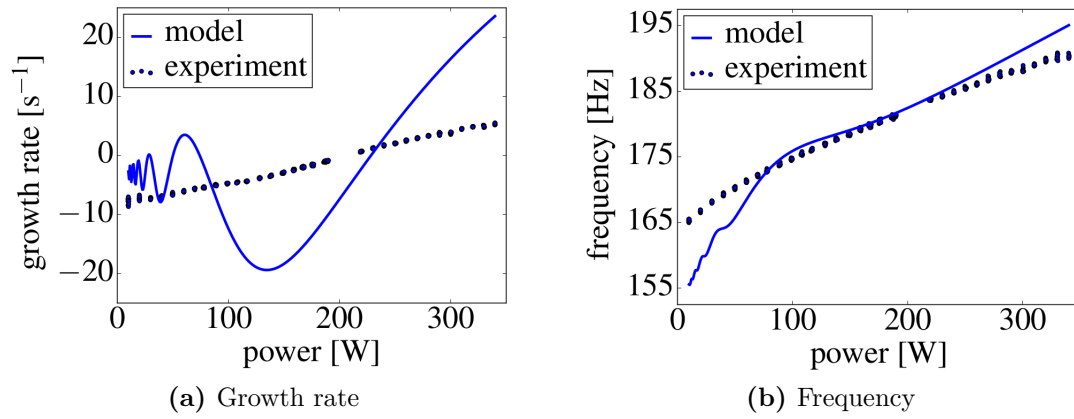


Figure 3.9: Experimental observations and model predictions of growth rate (left) and frequency (right) obtained by [34]. The reflection coefficient is treated as a regression parameter

Chapter 4

Conclusions and future work

The aim of the present study is to try to develop a novel methodology able to predict thermoacoustic oscillations over an entire range of operating conditions from a low order model containing the most important physics. This methodology is tested on a simple thermoacoustic system, namely an electrically-heated Rijke tube. Experiments are carried out at different powers, and gas and wall temperatures are measured at different locations along the tube. Because at these operating conditions the system is stable, an acoustic pulse is provided by a loudspeaker, and the decay of pressure oscillations is measured by six microphones placed along the tube. From these measurements, growth rate and frequencies of oscillations are extracted, and the reflection coefficient of the acoustic waves at the two ends of the tube is estimated using the multi-microphone method. This allows us to have many thousand experimental data.

We derive an unsteady 1D model with the goal of describing the buoyancy-driven base flow that sets up once the electric heater is switched on. This model depends on three parameters: an inner Nusselt number, an outer Nusselt number, and a pressure loss coefficient. Because these parameters are unknown, we infer them from experiments using Bayesian regression. An Ensemble Kalman Filter is developed so that experimental observations are assimilated into the model. Mean values and uncertainties are available as a result. Inlet gas velocity, mean temperatures upstream and downstream of the heater, and reflection coefficients estimated from experiments are fed into a low order thermoacoustic model. A network model is used, similarly to [10, 11]. The relationship between heat release and velocity fluctuations is provided by a simple $n - \tau$ model, and these two parameters are in turn modelled as $n = k_n q$, and $\tau = k_\tau (0.2 d_w / U_1)$. The thermoacoustic model outputs the predicted growth rate and frequency of the oscillations for any operating conditions, once the model parameters are known. The two model parameters (k_n and k_τ) are estimated by simple regression.

The growth rate predictions are shown to be very sensitive to the reflection coefficient. This is proved analytically for the no-heating limiting case, and numerically for the heating case, and therefore highlights that the model is not sufficiently robust to small fluctuations in the reflection coefficient, which, due to intrinsic noise, are always present if this parameter is estimated from experiments. This conclusion leads us to treat real and imaginary parts of the reflection coefficient as regression parameters. Very good agreement between experimental observations

and model predictions is found when this approach is followed.

For future work, we aim at providing the answer to several questions that at this stage are still unanswered. First of all, an important consideration is that our final goal is to predict the system behaviour using only the parameters estimated by the Ensemble Kalman Filter. In other words, when changing the configuration of the Rijke tube and re-running the base flow model, we aim at not performing any data assimilation. To check whether this approach is feasible, it is first necessary to re-run the base flow model in the same configuration without assimilating any data, using the parameters learned from the Ensemble Kalman Filter, so as to see how well the model performs when data are not assimilated.

Moreover, we aim at modelling the acoustic damping introduced by the two rods, which is absent at the moment. Because it is challenging to account for a distributed source of damping in a network model, a partial workaround would be to add a source of damping concentrated at the heater location, because this would only change one of the four entries of the eigenmatrix shown in Eq. (2.93). In addition to this, we want to test more elaborated models to describe the coupling between the heat release rate and velocity fluctuations, and compare their performance.

Another important point concerns the very high sensitivity of the thermoacoustic model to its parameters. Because of this, it seems that the methodology of learning the model parameters on a certain configuration and using them to make predictions on different configurations can only work if a probabilistic approach is used instead of simple regression. Building a probability density function of the model parameters would allow us to propagate the uncertainty in the predictions, so as to associate an error bar to the growth rate and frequency of the model.

Eventually, it might be interesting to report that the overall simulation takes approximately 12 h, and is therefore faster than the experiment, which lasts 20.4 h. The simulation time is further reduced when, in the future, no data assimilation will be performed. This is a positive feature of the model, as it allows for online modifications to the system. Indeed, if the overall approach gives positive results, it is possible to write a routine that takes as input the operating conditions of the system, predicts as output its stability/instability, and in case of instability tells how to modify the system on the run to make it stable by using adjoint-based sensitivity analysis.

Appendices

Base flow model routine

```
function [T, U] = fun_conj_heat(D1, D2, T_opt, parameters, ...
T_amb, time_interval, total_time)

% T: Temperature vector [K] (solid and gas) after time_interval
% seconds
% U: Inlet gas velocity [m/s] at timestep n+1
% D2: Fourth-order accurate differentiation matrix of
% first-order spatial derivatives using FDM with centered
% approximations everywhere, but near the boundaries
% D2: Fourth-order accurate differentiation matrix of
% second-order spatial derivatives using FDM with centered
% approximations everywhere, but near the boundaries
% T_opt: Temperature vector [K] (solid and gas) at initial time.
% This comes from the Kalman filter.
% parameters: model parameters (Nu_i, Nu_o and k_i). These come
% from the Kalman filter.
% T_amb: ambient temperature [K] after time_interval seconds
% time_interval: physical time [s] to run the model for
% total_time: physical total time [s] to keep track of the heat
% input to the system so as to reproduce the
% experiment

% Solve 1D Advection Diffusion equation with conjugate
% heat transfer and using FD matrices
%
% Equation for the solid:
% d(T_s)/dt = lam_1 * d2(T_s)/dx^2
% - lam_2 * Nu_o * T_s
% + lam_3 * Nu_i * (T_g - T_s)
%
% Equation for the gas
% d(T_g)/dt = (1+T_g) * (
% - U*d(T_g)/dx
% + lam_4 * d2(T_g)/dx^2
% - lam_5 * Nu_i * (T_g - T_s)
% + qdh
% )
%
% Boundary conditions for both solid and gas:
% Homogenous Neumann at x = 0 and +1

% Input dimensional variables
L_t = 1.0; % m tube length
d_t = 0.0474; % m inner diameter of tube
t_t = 0.0017; % m thickness of tube
g = 9.81; % m/s^2 gravitational acceleration
lam_s = 50; % W/m/K thermal conductivity of solid
lam_a = 0.026; % W/m/K thermal conductivity of ambient gas
c_s = 500; % J/kg/K specific heat capacity of solid
c_g = 1000; % J/kg/K specific heat capacity of gas
rho_s = 8000; % kg/m^3 density of solid
rho_a = 1.2; % kg/m^3 density of ambient gas
mu = 1.8e-5; % kg/m/s dynamic viscosity of gas

% Set the heat release rate as a function of time to reproduce
% the experiment dynamics, i.e. a step-function with jumps
% every 68 min. Enter total heat release rate as the
% envelope integrates to 1
qdh = 10 + 10*floor(total_time/60/68); % Watts

% Assign ambient temperature to T_a
T_a = T_amb; % K ambient temperature

% Derive more dimensional variables
pi_i = pi*d_t; % m inner perimeter of solid
pi_o = pi*(d_t + 2*t_t); % m outer perimeter of solid
A_s = pi*(t_t*d_t + t_t^2); % m^2 cross-section of solid
A_g = pi*d_t^2/4; % m^2 cross-section of gas

% Calculate dimensionless variables
lam_1 = lam_s/rho_s/c_s/L_t^1.5/g^0.5;
lam_2 = lam_a/rho_s/c_s*pi_o/A_s/g^0.5/L_t^0.5;
lam_3 = lam_a/rho_s/c_s*pi_i/A_s/g^0.5/L_t^0.5*L_t/d_t;
lam_4 = lam_a/rho_a/c_g/L_t^1.5/g^0.5;
lam_5 = lam_a/rho_a/c_g*pi_i/A_g/g^0.5/L_t^0.5*L_t/d_t;
qdh = L_t^0.5/g^0.5/T_a/rho_a/A_g/c_g*qdh;

% Set dimensionless parameters
didt = 1.0; % d_iris/d_tube
Nu_o_opt = parameters(1);
Nu_i_opt = parameters(2);
ki_opt = parameters(3);

% Set reference scales
L_ref = L_t;
t_ref = (L_t/g)^0.5;
T_ref = T_a;

% Set number of points (minus one) in the domain and compute
% grid parameters
x = linspace(0,+1,N+1)';
dx = 1/N;
m = dx*ones(N+1,1);

% Modification at the boundaries to correctly compute
% the values of the integrals
m(1) = dx/2; m(N+1) = dx/2;

% Create identity and null matrices
I = eye(N+1);
Z = zeros(N+1);

% Set timestep and number of timesteps to march for
dt = 0.05;
NT = round(time_interval/t_ref/dt);

% Set the heat release rate envelope v(x)
% and normalize it so that it integrates to 1
v = exp(-1000*(x-0.25).^2);
v = v/(m'*v);

% Write the governing equations in matrix form:
% dT/dt = ( A + (T_g+1) * B ) * T

% Create submatrices of A:
% A_ss contains terms affecting d(T_s)/dt that depend on T_s
A_ss = lam_1*D2 - lam_2*Nu_o_opt*I - lam_3*Nu_i_opt*I;
% A_gs contains terms affecting d(T_s)/dt that depend on T_g
A_gs = lam_3*Nu_i_opt*I;
% Assemble the A matrix
A = [A_ss, A_gs; Z, Z];

% Create submatrices of B:
% B_sg contains terms affecting d(T_g)/dt that depend on T_s
B_sg = lam_5*Nu_i_opt*I;
% B_gg0 contains terms affecting d(T_g)/dt that depend on T_g,
% but not the velocity. Thus B will be assembled in the loop
B_gg0 = lam_4*D2 - lam_5*Nu_i_opt*I;

% Assemble the source term vector
S = [zeros(N+1,1); qdh*v];

% Create the initial state vector in non-dimensional form
T = T_opt/T_ref - 1;

% Define variable to compute U in the first timestep
Told = T;

% March forward in time for NT timesteps using RK4
for nn = 1:NT

% Work out U from T
U = sqrt((m'*(T(N+2:2*N+2))./(1+T(N+2:2*N+2)))) ./ ...
((1+T(end))*didt.^-4 - 1 + ki_opt/2*(max(T(N+2:end))+1));
```

```

% Update the B matrix
B_gg = B_gg0 - U*D1;
% Assemble the B matrix
B = [Z , Z ; B_sg , B_gg];

% March forward in time with RK4
Told = T;
rhsSum = 0;
RHS = (A + diag(T+1)*B)*T + diag(T+1)*S;
T = Told + 0.5*dt*RHS;
rhsSum = RHS;
RHS = (A + diag(T+1)*B)*T + diag(T+1)*S;
T = Told + 0.5*dt*RHS;
rhsSum = rhsSum + 2*RHS;
RHS = (A + diag(T+1)*B)*T + diag(T+1)*S;
T = Told + dt*RHS;
rhsSum = rhsSum + 2*RHS;
RHS = (A + diag(T+1)*B)*T + diag(T+1)*S;

rhsSum = rhsSum + RHS;
T = Told + dt*rhsSum/6;

% Set the solid boundary condition at x = 0
T(1) = T(1) - D1(1,:)*T(1:N+1) / D1(1,1);
% Set the solid boundary condition at x = +1
T(N+1) = T(N+1) - D1(N+1,:)*T(1:N+1) / D1(N+1,N+1);
% Set the gas boundary condition at x = 0
T(N+2) = T(N+2) - D1(1,:)*T(N+2:2*N+2) / D1(1,1);
% Set the gas boundary condition at x = +1
T(2*N+2) = T(2*N+2) - D1(N+1,:)*T(N+2:2*N+2) / D1(N+1,N+1);

end

% Switch to dimensional values
T = T_ref*(1+T);
U = U*sqrt(g*L_t);

end

```

Ensemble Kalman Filter routine

```
#!/usr/bin/env python3

import numpy as np
import numpy.linalg as la
import numpy.random as random
from mpi4py import MPI

import plinalg as pla

class Kalman:
    def __init__(self):
        # mpi4py.MPI
        self.comm = MPI.COMM_WORLD
        self.myid = self.comm.Get_rank()
        self.numprocs = self.comm.Get_size()

        # numpy.random
        random.seed(self.myid + 1)

    def filter(self, psi_f, meas, locs, s_meas):
        comm = self.comm
        myid = self.myid
        numprocs = self.numprocs

        if type(s_meas) == float:
            s_meas = np.full(len(locs), s_meas)
            if not s_meas.size == len(locs):
                raise ValueError()

        psi_f_mean = np.empty_like(psi_f)
        comm.Allreduce(psi_f, psi_f_mean, MPI.SUM)
        psi_f_mean /= numprocs
        psi_f_prime = psi_f - psi_f_mean

        # Psi_f_meas.
        if myid == 0:
            Psi_f_meas = np.empty((numprocs, meas.size))
        else:
            Psi_f_meas = []
        comm.Gatherv(psi_f_prime[locs], Psi_f_meas)

        if myid == 0:
            # A.
            A = np.matmul(Psi_f_meas.T, Psi_f_meas)
            A[np.diag_indices_from(A)] +=
                (numprocs - 1) * s_meas * s_meas

            # B.
            B = np.empty_like(Psi_f_meas)
            for i in range(numprocs):
                B[i, :] = meas - psi_f_mean[locs]
            B += s_meas * random.standard_normal(B.shape) -
                Psi_f_meas

            # Solve AX = B.
            B = la.lstsq(A, B.T, rcond=None)[0]

            # Matrix multiplication.
            B = np.matmul(Psi_f_meas, B)
        else:
            B = np.empty((numprocs, numprocs))

        # Parallel matrix multiplication.
        comm.Bcast(B)
        Psi_f = pla.col2block(psi_f, psi_f.size, numprocs, comm)
        Delta_Psi = np.matmul(Psi_f.T, B).T
        delta_psi =
            pla.block2col(Delta_Psi, psi_f.size, numprocs, comm)
        psi_a = psi_f + delta_psi

        return psi_a

    def smooth(self, psi_f, psi_a, psi_s):
        comm = self.comm
        myid = self.myid
        numprocs = self.numprocs

        # (psi^f)_{k+1}
        psi_f_mean = np.empty_like(psi_f)
        comm.Allreduce(psi_f, psi_f_mean, MPI.SUM)
        psi_f_mean /= numprocs
        psi_f_prime = psi_f - psi_f_mean
        Psi_f =
            pla.col2block
            (psi_f_prime, psi_f_prime.size, numprocs, comm)

        # (psi^a)_k
        psi_a_mean = np.empty_like(psi_a)
        comm.Allreduce(psi_a, psi_a_mean, MPI.SUM)
        psi_a_mean /= numprocs
        psi_a_prime = psi_a - psi_a_mean
        Psi_a = pla.col2block(psi_a_prime,

psi_a_prime.size, numprocs, comm)

        # (psi^s)_{k+1}
        psi_s_mean = np.empty_like(psi_s)
        comm.Allreduce(psi_s, psi_s_mean, MPI.SUM)
        psi_s_mean /= numprocs
        psi_s_prime = psi_s - psi_s_mean
        Psi_s =
            pla.col2block(psi_s_prime,
                psi_s_prime.size, numprocs, comm)

        B = Psi_s - Psi_f
        B = pla.matmul(Psi_f, B.T, comm)
        A = pla.matmul(Psi_f, Psi_f.T, comm)
        if myid == 0:
            C = la.lstsq(A, B)[0]
        else:
            C = np.empty((numprocs, numprocs))
        comm.Bcast(C)
        delta_Psi = np.matmul(Psi_a.T, C).T
        delta_psi =
            pla.block2col(delta_Psi, psi_a.size, numprocs, comm)
        psi = psi_a + delta_psi

        return psi

if __name__ == "__main__":
    import string
    import matplotlib.pyplot as plt

    no_pts = 10
    s_meas = 1.
    no_meas = 10

    comm = MPI.COMM_WORLD
    myid = comm.Get_rank()
    numprocs = comm.Get_size()

    random.seed(numprocs + myid + 1)

    # Kalman filter.
    psi_f = np.zeros(no_pts) + random.randn(no_pts)
    meas = np.zeros(no_meas) + s_meas * random.randn(no_meas)
    comm.Bcast(meas)
    locs = range(0, no_pts, int(no_pts/no_meas))

    kal = Kalman()
    psi_a = kal.filter(psi_f, meas, locs, s_meas)

    # Analysis.
    psi_f_mean = np.empty_like(psi_f)
    comm.Allreduce(psi_f, psi_f_mean, MPI.SUM)
    psi_f_mean /= numprocs

    psi_f_prime = psi_f - psi_f_mean
    psi_f_cov = np.outer(psi_f_prime, psi_f_prime)
    if myid == 0:
        comm.Reduce(MPI.IN_PLACE, psi_f_cov)
    else:
        comm.Reduce(psi_f_cov, psi_f_cov)
    psi_f_cov /= numprocs - 1

    psi_a_mean = np.empty_like(psi_a)
    comm.Allreduce(psi_a, psi_a_mean, MPI.SUM)
    psi_a_mean /= numprocs

    psi_a_prime = psi_a - psi_a_mean
    psi_a_cov = np.outer(psi_a_prime, psi_a_prime)
    if myid == 0:
        comm.Reduce(MPI.IN_PLACE, psi_a_cov)
    else:
        comm.Reduce(psi_a_cov, psi_a_cov)
    psi_a_cov /= numprocs - 1

    if myid == 0:
        f = open("kalman.log", 'w')

        # Mean.
        fig = plt.figure()
        ax = fig.gca()
        ax.plot(psi_f_mean, label='forecast')
        ax.plot(psi_a_mean, label='analysis')
        ax.plot(locs, meas, label='measurements')
        ax.legend()
        fig.savefig("kalman_mean.eps", format='eps')

        f.write("Mean:\n")
        for i in range(psi_a_mean.size):
            f.write("{: f} ".format(psi_a_mean[i]))
        f.write('\n')
        f.write('\n')
```

```

# Covariance.
fig = plt.figure()
ax = fig.gca()
m = ax.pcolor(psi_a_cov)
fig.colorbar(m)
fig.savefig("kalman_cov.eps", format='eps')

f.write("Covariance:\n")
for i in range(psi_a_cov.shape[0]):
    for j in range(psi_a_cov.shape[1]):
        f.write("{: f} ".format(psi_a_cov[i][j]))
        f.write('\n')
    f.write('\n')

print(np.mean(np.diagonal(psi_a_cov)))

f.close()

# Kalman filter.
meas_1 = np.zeros(no_meas) + s_meas * random.randn(no_meas)
comm.Bcast(meas_1)
psi_a_1 = kal.filter(psi_a, meas_1, locs, s_meas)

# Analysis.
psi_f_mean = np.empty_like(psi_a)
comm.Allreduce(psi_a, psi_f_mean, MPI.SUM)
psi_f_mean /= numprocs

psi_f_prime = psi_a - psi_f_mean
psi_f_cov = np.outer(psi_f_prime, psi_f_prime)
if myid == 0:
    comm.Reduce(MPI.IN_PLACE, psi_f_cov)
else:
    comm.Reduce(psi_f_cov, psi_f_cov)
psi_f_cov /= numprocs - 1

psi_a_mean = np.empty_like(psi_a_1)
comm.Allreduce(psi_a_1, psi_a_mean, MPI.SUM)
psi_a_mean /= numprocs

psi_a_prime = psi_a_1 - psi_a_mean
psi_a_cov = np.outer(psi_a_prime, psi_a_prime)
if myid == 0:
    comm.Reduce(MPI.IN_PLACE, psi_a_cov)
else:
    comm.Reduce(psi_a_cov, psi_a_cov)
psi_a_cov /= numprocs - 1

if myid == 0:
    f = open("kalman_1.log", 'w')

    # Mean.
    fig = plt.figure()
    ax = fig.gca()
    ax.plot(psi_f_mean, label='forecast')
    ax.plot(psi_a_mean, label='analysis')
    ax.plot(locs, meas_1, label='measurements')
    ax.legend()
    fig.savefig("kalman_mean_1.eps", format='eps')

    f.write("Mean:\n")
    for i in range(psi_a_mean.size):
        f.write("{: f} ".format(psi_a_mean[i]))
        f.write('\n')
    f.write('\n')

    # Covariance.
    fig = plt.figure()
    ax = fig.gca()
    m = ax.pcolor(psi_a_cov)
    fig.colorbar(m)
    fig.savefig("kalman_cov_1.eps", format='eps')

    f.write("Covariance:\n")
    for i in range(psi_a_cov.shape[0]):
        for j in range(psi_a_cov.shape[1]):
            f.write("{: f} ".format(psi_a_cov[i][j]))
            f.write('\n')
        f.write('\n')

    print(np.mean(np.diagonal(psi_a_cov)))

    f.close()

        f.write("{: f} ".format(psi_a_cov[i][j]))
        f.write('\n')
        f.write('\n')

        print(np.mean(np.diagonal(psi_a_cov)))

        f.close()

# Smoother
psi_s = kal.smooth(psi_a, psi_a_1, psi_a_1)

# Analysis.
psi_f_mean = np.empty_like(psi_a)
comm.Allreduce(psi_a, psi_f_mean, MPI.SUM)
psi_f_mean /= numprocs

psi_f_prime = psi_a - psi_f_mean
psi_f_cov = np.outer(psi_f_prime, psi_f_prime)
if myid == 0:
    comm.Reduce(MPI.IN_PLACE, psi_f_cov)
else:
    comm.Reduce(psi_f_cov, psi_f_cov)
psi_f_cov /= numprocs - 1

psi_a_mean = np.empty_like(psi_s)
comm.Allreduce(psi_s, psi_a_mean, MPI.SUM)
psi_a_mean /= numprocs

psi_a_prime = psi_s - psi_a_mean
psi_a_cov = np.outer(psi_a_prime, psi_a_prime)
if myid == 0:
    comm.Reduce(MPI.IN_PLACE, psi_a_cov)
else:
    comm.Reduce(psi_a_cov, psi_a_cov)
psi_a_cov /= numprocs - 1

psi_s_mean = np.empty_like(psi_a_1)
comm.Allreduce(psi_a_1, psi_s_mean, MPI.SUM)
psi_s_mean /= numprocs

if myid == 0:
    f = open("kalman_2.log", 'w')

    # Mean.
    fig = plt.figure()
    ax = fig.gca()
    ax.plot(psi_f_mean, label='filtered')
    ax.plot(psi_a_mean, label='smoothed')
    ax.plot(psi_s_mean, label='smoothed')
    ax.legend()
    fig.savefig("kalman_mean_2.eps", format='eps')

    f.write("Mean:\n")
    for i in range(psi_a_mean.size):
        f.write("{: f} ".format(psi_a_mean[i]))
        f.write('\n')
    f.write('\n')

    # Covariance.
    fig = plt.figure()
    ax = fig.gca()
    m = ax.pcolor(psi_a_cov)
    fig.colorbar(m)
    fig.savefig("kalman_cov_2.eps", format='eps')

    f.write("Covariance:\n")
    for i in range(psi_a_cov.shape[0]):
        for j in range(psi_a_cov.shape[1]):
            f.write("{: f} ".format(psi_a_cov[i][j]))
            f.write('\n')
        f.write('\n')

    print(np.mean(np.diagonal(psi_a_cov)))

    f.close()

```

Experimental decay rate and frequency routine

```

function [t,p_raw,Hampl,Hfreq,coeff,s_real,s_imag] =
fun_DecayRate(f_exc,Fs,microphone_number,V,lower_limit,
upper_limit)

% Calibration valuesfrequencies in the first column, decibels
% in the second one
calibr1 = load('E:\Experiments\calibrationData\263705.dat');
calibr2 = load('E:\Experiments\calibrationData\263706.dat');
calibr3 = load('E:\Experiments\calibrationData\263707.dat');
calibr4 = load('E:\Experiments\calibrationData\263703.dat');
calibr5 = load('E:\Experiments\calibrationData\263702.dat');

% Conversion factors for the microphones
V_to_Pa_1 = 2.99/1000;
V_to_Pa_2 = 2.95/1000;
V_to_Pa_3 = 2.85/1000;
V_to_Pa_4 = 3.02/1000;
V_to_Pa_5 = 2.80/1000;

% Apply calibration and convert voltage fluctuations into
% pressure fluctuations
switch(microphone_number)
case{1}
V = V - mean(V);
V = ...
V/10^(0.05*interp1(calibr1(:,1),calibr1(:,2),f_exc));
p_raw = V/V_to_Pa_1;
case{2}
V = V - mean(V);
V = ...
V/10^(0.05*interp1(calibr2(:,1),calibr2(:,2),f_exc));
p_raw = V/V_to_Pa_2;
case{3}
V = V - mean(V);
V = ...
V/10^(0.05*interp1(calibr3(:,1),calibr3(:,2),f_exc));
p_raw = V/V_to_Pa_3;
case{4}
V = V - mean(V);
V = ...
V/10^(0.05*interp1(calibr4(:,1),calibr4(:,2),f_exc));
p_raw = V/V_to_Pa_4;
case{5}
V = V - mean(V);
V = ...
V/10^(0.05*interp1(calibr5(:,1),calibr5(:,2),f_exc));
p_raw = V/V_to_Pa_5;
end

% Apply a Butter filter to remove too high/low frequencies
Fc = f_exc; % center frequency
dFc = 25; % delta F
[B,A]=butter(3,[Fc-dFc Fc+dFc]/(Fs/2),'bandpass');
p_raw = filtfilt(B,A,p_raw);

% Compute Fast Fourier Transform
ndata = length(p_raw);
NFFT = 2^nextpow2(ndata);

% Compute frequency domain. The highest captured frequency
% is Fs/2. Above this, frequencies are aliased
f = Fs/2*linspace(0,1,NFFT/2+1);
dt = 1/Fs;

% Compute the Fourier transform in the two-sided spectrum
p_fourierTransf = fft(p_raw,NFFT)*dt;
p_fourierTransf = p_fourierTransf(1:NFFT/2+1);
p_fourierTransf(2:end-1) = 2*p_fourierTransf(2:end-1);
freq = ...
f(find( abs(p_fourierTransf) == max(abs(p_fourierTransf)))));

% Calculate Hilbert Transform
% Note: multiply by Fs correponds to divide by dt
Philb = hilbert(p_raw);
Hampl = abs(Philb);
Hphase = unwrap(angle(Philb));
Hfreq = diff(unwrap([Hphase ; Hphase(end)]))*Fs/(2*pi);

N = length(p_raw);
t = [0:N-1]./Fs;
t = t';

coeff = polyfit(t(t>lower_limit & t<upper_limit), ...
log(Hampl(t>lower_limit & t<upper_limit)),1);

s_real = coeff(1);
s_imag = freq;

end

```


Multi-microphone method routine

```

function [R] = fun_MMM(i,f_exc,Fs,L,T_avg,x,V)

% This function makes use of the multi-mic method to
% compute at timestep i the reflection coefficient R
% at the upper end of a Rijke tube with length L, given
% the sampling frequency Fs, the average temperature in
% the relevant section of the tube T_avg, the position
% of the five microphone x, and the voltage signals
% recorded by the microphone.
%
% Note 1: f_exc is the frequency at which the tube is
% excited, which is needed only to properly choose the
% calibration factor.
%
% Note 2: the vector x must contain the distance of each
% microphone from the bottom end, starting from the
% top microphone.
%
% Note 3: the first column of V must contain the voltage
% signal of the top-most microphone, whereas the last
% column the voltage signal of the bottom-most one.
%
% Note 4: the microphone serial numbers must be the
% following ones (from top to bottom):
% 263705, 263706, 263707, 263703, 263702.

% Compute the average speed of sound
gamma = 1.4;
Rgas = 8314.5;
Mm = 28.964;
c = sqrt(gamma*(Rgas/Mm)*T_avg);

% Position of the microphones
x1_base = x(1); % Distance from base of the tube
x2_base = x(2); % Distance from base of the tube
x3_base = x(3); % Distance from base of the tube
x4_base = x(4); % Distance from base of the tube
x5_base = x(5); % Distance from base of the tube
x1 = L - x1_base; % Distance from outlet of the tube (>0)
x2 = L - x2_base; % Distance from outlet of the tube (>0)
x3 = L - x3_base; % Distance from outlet of the tube (>0)
x4 = L - x4_base; % Distance from outlet of the tube (>0)
x5 = L - x5_base; % Distance from outlet of the tube (>0)

% Calibration values - frequencies in the first column,
% decibels in the second one
calibr1 = load('E:\Experiments\calibrationData\263705.dat');
calibr2 = load('E:\Experiments\calibrationData\263706.dat');
calibr3 = load('E:\Experiments\calibrationData\263707.dat');
calibr4 = load('E:\Experiments\calibrationData\263703.dat');
calibr5 = load('E:\Experiments\calibrationData\263702.dat');

% Conversion factors for the microphones
V_to_Pa_1 = 2.99/1000;
V_to_Pa_2 = 2.95/1000;
V_to_Pa_3 = 2.85/1000;
V_to_Pa_4 = 3.02/1000;
V_to_Pa_5 = 2.80/1000;

% Define voltage vectors
V1 = V(:,1);
V2 = V(:,2);
V3 = V(:,3);
V4 = V(:,4);
V5 = V(:,5);

% Apply calibration
V1 = V1/10^(0.05*interp1(calibr1(:,1),calibr1(:,2),f_exc));
V2 = V2/10^(0.05*interp1(calibr2(:,1),calibr2(:,2),f_exc));
V3 = V3/10^(0.05*interp1(calibr3(:,1),calibr3(:,2),f_exc));
V4 = V4/10^(0.05*interp1(calibr4(:,1),calibr4(:,2),f_exc));
V5 = V5/10^(0.05*interp1(calibr5(:,1),calibr5(:,2),f_exc));

% Subtract the mean value so that the amplitude of the
% zero-frequency signal is zero. This is done because
% otherwise we could get that the zero-frequency signal
% is stronger than the signal at the excitation frequency
V1_prime = V1 - mean(V1);
V2_prime = V2 - mean(V2);
V3_prime = V3 - mean(V3);
V4_prime = V4 - mean(V4);
V5_prime = V5 - mean(V5);

% Convert voltage fluctuations into pressure fluctuations
p_raw1 = V1_prime/V_to_Pa_1;
p_raw2 = V2_prime/V_to_Pa_2;
p_raw3 = V3_prime/V_to_Pa_3;
p_raw4 = V4_prime/V_to_Pa_4;
p_raw5 = V5_prime/V_to_Pa_5;

% Compute time interval between two samples and total
% number of data
dt = 1/Fs;
ndata = length(V1);

% Compute coefficient to improve performance of FFT
NFFT = 2^nextpow2(ndata);

% Compute frequency domain. The highest captured frequency
% is Fs/2. Above this, frequencies are aliased
f = Fs/2*linspace(0,1,NFFT/2+1);

% Compute the Fourier transform in the two-sided spectrum
p1_fourTransf = fft(p_raw1,NFFT)*dt;
p2_fourTransf = fft(p_raw2,NFFT)*dt;
p3_fourTransf = fft(p_raw3,NFFT)*dt;
p4_fourTransf = fft(p_raw4,NFFT)*dt;
p5_fourTransf = fft(p_raw5,NFFT)*dt;

% Compute the Fourier transform in the one-sided spectrum,
% i.e. taking into account only frequencies up to Fs/2.
% The factor 2 is needed in order not to change the energy
% content of the spectrum
p1_fourTransf = p1_fourTransf(1:NFFT/2+1);
p1_fourTransf(2:end-1) = 2*p1_fourTransf(2:end-1);
p2_fourTransf = p2_fourTransf(1:NFFT/2+1);
p2_fourTransf(2:end-1) = 2*p2_fourTransf(2:end-1);
p3_fourTransf = p3_fourTransf(1:NFFT/2+1);
p3_fourTransf(2:end-1) = 2*p3_fourTransf(2:end-1);
p4_fourTransf = p4_fourTransf(1:NFFT/2+1);
p4_fourTransf(2:end-1) = 2*p4_fourTransf(2:end-1);
p5_fourTransf = p5_fourTransf(1:NFFT/2+1);
p5_fourTransf(2:end-1) = 2*p5_fourTransf(2:end-1);

% Filtering
p1_fourTransf(f < 100 | f > 300) = 0 + 0*1i;
p2_fourTransf(f < 100 | f > 300) = 0 + 0*1i;
p3_fourTransf(f < 100 | f > 300) = 0 + 0*1i;
p4_fourTransf(f < 100 | f > 300) = 0 + 0*1i;
p5_fourTransf(f < 100 | f > 300) = 0 + 0*1i;

% Find excitation frequency
f1_exc = f(find(abs(p1_fourTransf) == max(abs(p1_fourTransf))));
f2_exc = f(find(abs(p2_fourTransf) == max(abs(p2_fourTransf))));
f3_exc = f(find(abs(p3_fourTransf) == max(abs(p3_fourTransf))));
f4_exc = f(find(abs(p4_fourTransf) == max(abs(p4_fourTransf))));
f5_exc = f(find(abs(p5_fourTransf) == max(abs(p5_fourTransf))));

% Check that the excitation frequencies are not that different.
% Criterion: index may differ of at most 1 position
if abs(find(f == f1_exc) - find(f == f2_exc)) > 1 || ...
abs(find(f == f1_exc) - find(f == f3_exc)) > 1 || ...
abs(find(f == f1_exc) - find(f == f4_exc)) > 1 || ...
abs(find(f == f1_exc) - find(f == f5_exc)) > 1 || ...
abs(find(f == f1_exc) - find(f == f6_exc)) > 1
fprintf('Excitation frequencies different when i = %d.\n ...
f1_exc = %f , f2_exc = %f , f3_exc = %f , ...
f4_exc = %f , f5_exc = %f , f6_exc = %f \n', ...
i,f1_exc,f2_exc,f3_exc,f4_exc,f5_exc,f6_exc);
return
end

% Compute pressure values at excitation frequency
p1_exc = p1_fourTransf( find(f == f1_exc) );
p2_exc = p2_fourTransf( find(f == f1_exc) );
p3_exc = p3_fourTransf( find(f == f1_exc) );
p4_exc = p4_fourTransf( find(f == f1_exc) );
p5_exc = p5_fourTransf( find(f == f1_exc) );

% Compute the wavenumbers.
% Set the Mach number to zero for no flow
M = 0;
k = 2*pi*f1_exc/c;
k_incident = k/(1+M);
k_reflected = k/(1-M);

% Compute the amplitudes of the incident and reflected pressure
% signal in the frequency domain, respectively. This comes from
% solving numerically Ax=b with a least-square method, where
% A = [exp(1i*x1*k_incident) exp(-1i*x1*k_reflected); ...
% exp(1i*x2*k_incident) exp(-1i*x2*k_reflected)];
% b = [p1_exc ; p2_exc];
% x = [A_coeff ; B_coeff];
% NOTE: Here x are positive values!
matrix = [exp(1i*x1*k_incident) exp(-1i*x1*k_reflected); ...
exp(1i*x2*k_incident) exp(-1i*x2*k_reflected); ...
exp(1i*x3*k_incident) exp(-1i*x3*k_reflected); ...
exp(1i*x4*k_incident) exp(-1i*x4*k_reflected); ...
exp(1i*x5*k_incident) exp(-1i*x5*k_reflected); ...
exp(1i*x6*k_incident) exp(-1i*x6*k_reflected)];
RHS = [p1_exc; p2_exc; p3_exc; p4_exc; p5_exc; p6_exc];

% Use least-square method for undetermined linear system of

```

```
% equations
sol = matrix\RHS;
A_coeff = sol(1);
B_coeff = sol(2);

% Compute the reflection coefficients for top end,
R = B_coeff/A_coeff;
```

Newton's method routine

```

function s = NewtonMethod(params)

% Define geometry
L = 1;           % tube length [m]
d = 0.0474;     % tube diameter [m]
dw = 0.559e-3; % wire diameter [m]
x_h = 0.25;     % position of the heater [m]

% Define inlet pressure
p1 = 101325;    % ambient pressure [m]

% Define constants for air
gamma = 1.4;    % specific heat capacity ratio for air
Rgas = 8314.5; % universal gas constant [J/kmol/K]
Mm = 28.964;   % molar mass [kg/kmol]

% Extract parameters
k_n = params(1);
k_tau = params(2);

% Compute the determinant of the eigenvalue problem matrix,
% which consists of the following four terms:
L11 = sym('(1 + Ru*exp(-s_model*tau_u))');
L12 = sym('(1 + Rd*exp(-s_model*tau_d))');
L21 = sym('(1 - Ru*exp(-s_model*tau_u))* ...
           (1 + (gamma - 1)/gamma/p1*n*exp(-s_model*tau))');
L22 = sym('(c2/c1*(1 - Rd*exp(-s_model*tau_d))');

% Compute det(A) and its first derivative with respect
% to the eigenvalue
f = L11.*L22 - L21.*L12;
df_ds = diff(f,'s_model');

% Define number of experiments
n_exp = 1080;

% Initialise residual function
s = zeros(1,n_exp);

for i = 1:n_exp

    load('regressionVariables.mat');
    load('RHS.mat');

    T1 = T1_avg(i);
    T2 = T2_avg(i);

    % Compute the average speed of sound in the upstream
    % and downstream parts of the tube, i.e. c1 and c2,
    % respectively
    c1 = sqrt(gamma*(Rgas/Mm)*T1_avg(i));
    c2 = sqrt(gamma*(Rgas/Mm)*T2_avg(i));

    % Define the time for the wave to travel back and
    % forth from the the heater to the bottom end, tau_u,
    % and from the heater to the top end, tau_d.
    tau_u = 2*x_h/c1;
    tau_d = 2*(L - x_h)/c2;

    % Define n and tau according to your model
    n = k_n*q(i);
    tau = k_tau*0.2*dw/ui(i);

    Ru = Ru(i);
    Rd = Rd(i);
    u1 = u1(i);

    % Newton's method
    n_iter = 0;
    tol = 1e-12;
    s_model = -9.6 + 2*pi*170*i;
    while abs(eval(f)) > tol
        s1 = s_model - eval(f)/eval(df_ds);
        s_model = s1;
        n_iter = n_iter + 1;
    end
    s(i) = s_model;
end
end
end

```


References

- [1] B. Higgins, On the sound produced by a current of hydrogen gas passing through a tube, *Journal of Natural Philosophy, Chemistry and the Arts* 1 (1802) 129–131.
- [2] J. W. Rayleigh, The explanation of certain acoustical phenomena, *Nature* 18 (1878) 319–321.
- [3] A. A. Putnam, W. R. Dennis, Burner oscillations of the gauze-tone type, *The Journal of the Acoustical Society of America* 26 (1954) 716–725.
- [4] B.-T. Chu, On the energy transfer to small disturbances in fluid flow (part i), *Acta Mechanica* 1 (1965) 215–234.
- [5] L. Crocco, Research on combustion instability in liquid propellant rockets, *Symposium (International) on Combustion* 12 (1969) 85 – 99.
- [6] J. C. Oefelein, V. Yang, Comprehensive review of liquid-propellant combustion instabilities in f-1 engines, *Journal of Propulsion and Power* 9 (1993) 657–677.
- [7] S. M. Candel, Combustion instabilities coupled by pressure waves and their active control, *Symposium (International) on Combustion* 24 (1992).
- [8] S. M. Candel, Combustion dynamics and control: progress and challenges, *Proceedings of the Combustion Institute* 29 (2002) 1–28.
- [9] L. Selle, G. Lartigue, T. Poinsot, R. Koch, K.-U. Schildmacher, W. Krebs, B. Prade, P. Kaufmann, D. Veynante, Compressible large eddy simulation of turbulent combustion in complex geometry on unstructured meshes, *Combustion and Flame* 137 (2004) 489 – 505.
- [10] S. R. Stow, A. P. Dowling, Thermoacoustic oscillations in an annular combustor, *Proceedings of ASME Turbo Expo* (2001).
- [11] J. G. Aguilar, L. Magri, M. P. Juniper, Adjoint-based sensitivity analysis of low-order thermoacoustic networks using a wave-based approach, *Journal of Computational Physics* 341 (2017) 163 – 181.
- [12] M. Summerfield, A theory of unstable combustion in liquid propellant rocket systems, *Journal of the American Rocket Society* 21 (1951) 108–114.
- [13] M. P. Juniper, R. Sujith, Sensitivity and nonlinearity of thermoacoustic oscillations, *Annual Review of Fluid Mechanics* 50 (2018) 661–689.

- [14] A. Avdonin, S. Jaensch, C. F. Silva, M. Češnovar, W. Polifke, Uncertainty quantification and sensitivity analysis of thermoacoustic stability with non-intrusive polynomial chaos expansion, *Combustion and Flame* 189 (2018) 300–310.
- [15] A. Tarantola, *Inverse problem theory and methods for model parameter estimation*, Society for Industrial and Applied Mathematics, 2005.
- [16] G. Evensen, *Data Assimilation: The Ensemble Kalman Filter*, 2nd ed., Springer, 2009.
- [17] P. L. Rijke, On the vibration of the air in a tube open at both ends, *Philosophical Magazine* 17 (1859) 419–422.
- [18] L. Magri, M. P. Juniper, Sensitivity analysis of a time-delayed thermo-acoustic system via an adjoint-based approach, *Journal of Fluid Mechanics* 719 (2013) 183–202.
- [19] J. H. Ferziger, M. Peric, *Computational Methods for Fluid Dynamics*, 3rd ed., Springer-Verlag Berlin Heidelberg, 2002, chapter 3, pp. 39–69.
- [20] B. Fornberg, Generation of finite difference formulas on arbitrarily spaced grids, *Mathematics of Computation* 51 (1988) 699–706.
- [21] B. Fornberg, Calculation of weights in finite difference formulas, *SIAM Review* 40 (1998) 685–691.
- [22] T. Saito, Vibrations of air-columns excited by heat supply, *The Japan Society of Mechanical Engineers* 8 (1965) 651–659.
- [23] G. Rigas, N. P. Jamieson, L. K. B. Li, M. P. Juniper, Experimental sensitivity analysis and control of thermoacoustic systems, *Journal of Fluid Mechanics* 787 (2016) R1.
- [24] N. P. Jamieson, G. Rigas, M. P. Juniper, Experimental sensitivity analysis via a secondary heat source in an oscillating thermoacoustic system, *International Journal of Spray and Combustion Dynamics* 9 (2017) 230–240.
- [25] N. P. Jamieson, M. P. Juniper, Experimental sensitivity analysis of a linearly stable thermoacoustic system via a pulsed forcing technique, *Experiments in Fluids* 58 (2017) 123.
- [26] N. P. Jamieson, M. P. Juniper, Experimental sensitivity analysis and the equivalence of pulsed forcing and feedback control in thermoacoustic systems, *Proceedings of ASME Turbo Expo* (2017).
- [27] H. Levine, J. Schwinger, On the radiation of sound from an unflanged circular pipe, *Physical Review* 73 (1948) 383–406.
- [28] M. Schumm, E. Berger, P. A. Monkewitz, Self-excited oscillations in the wake of two-dimensional bluff bodies and their control, *Journal of Fluid Mechanics* 271 (1994) 17–53.

-
- [29] A. F. Seybert, D. F. Ross, Experimental determination of acoustic properties using a two-microphone random-excitation technique, *The Journal of the Acoustical Society of America* 61 (1977) 1362–1370.
- [30] H. W. Liepmann, A. Roshko, *Elements of Gasdynamics*, Dover Publications, 2001, chapter 3, pp. 62 – 83.
- [31] F. Nicoud, K. Wieczorek, About the zero mach number assumption in the calculation of thermoacoustic instabilities, *International Journal of Spray and Combustion Dynamics* 1 (2009) 67–112.
- [32] M. J. Lighthill, The response of laminar skin friction and heat transfer to fluctuations in the stream velocity, *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 224 (1954) 1–23.
- [33] M. S. Bhatti, R. K. Shah (Eds.), *Handbook of Single-Phase Convective Heat Transfer*, New York: Wiley, 1987.
- [34] H. Yu, Inverse uncertainty quantification in thermoacoustics, 2017. PhD First-Year Report (Unpublished).