

User Guide for OSCILOS_SIM

Simulating combustion instabilities using Simulink

A. S. Morgans, a.morgans@imperial.ac.uk & M. Haas-Heger, mkh2149@columbia.edu, 14/12/2015

Overview of modelling methodology

Combustion instabilities, also known as thermoacoustic instabilities, are caused by a two-way coupling between acoustic waves and unsteady heat release rate. They are associated with successively growing oscillation amplitudes, which eventually saturate into a nonlinear state, most commonly limit cycle oscillations. For a detailed explanation of combustion instabilities and their modelling, along with relevant references, refer to the OSCILOS_Long Technical report, available at www.oscilos.com.

For simple combustors which can be represented as a heat source within a constant area duct, simulating the time evolution of thermoacoustic oscillations is possible using Simulink. The OSCILOS_SIM Simulink library contains the building blocks for this. The assumed combustor set-up consists of a heat source (typically a flame) within a constant area duct, through which a one-dimensional low Mach number mean flow passes, as shown in Fig. 1. The fluid either side of the heat source behaves as a perfect gas, with the specific heats considered constant throughout the duct. For a duct which is open at both ends, this is often referred to as a Rijke tube.

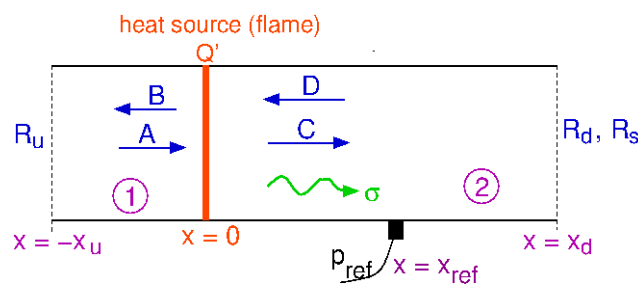


Figure 1: Schematic of a ducted flame combustor.

Simulations are restricted to “low” frequencies, where the flame is acoustically compact and can be represented as an infinitely thin sheet, and acoustic waves are one-dimensional plane waves. Unsteady heat release from the flame generates both plane acoustic waves and entropy waves (the latter are essentially temperature fluctuations). Their amplitudes are small enough for them to behave linearly with respect to the mean flow. Acoustic waves with strengths B and C then propagate away from the flame at the speed of sound relative to the flow. Their reflection from the duct ends is captured with pressure reflection coefficients R_u and R_d ; a variety of acoustic boundary conditions can be chosen. The entropy wave, σ , advects downstream of the heat source with the mean flow. It is silent within the duct, but if the downstream duct boundary represents a flow acceleration, “entropy noise” may be generated, and an acoustic wave will propagate back upstream towards the heat source. This is captured via an entropy reflection coefficient, R_s . The overall reflected incoming acoustic waves then have strengths A and D .

The unsteady heat release rate from the flame is assumed to respond to the acoustic velocity fluctuation just upstream of the flame i.e. $Q'/\bar{Q} = f(u'_1/\bar{u}_1)$. This flame model may be linear or nonlinear; only nonlinear flame models can capture saturation of instabilities into limit cycle oscillations.

Combustion instabilities can be suppressed using active feedback control. In a simple duct combustor, this is most commonly achieved by measuring the pressure fluctuation, p_{ref} , at location $x = x_{ref}$ within the duct, passing this through a real-time control algorithm, and sending the control output signal to an actuator. The actuator can be a fuel valve which modulates the fuel flow rate and hence the equivalence ratio, or a loudspeaker which generates an additional acoustic wave component within the duct. In the set-up above, fuel flow rate actuation is assumed.

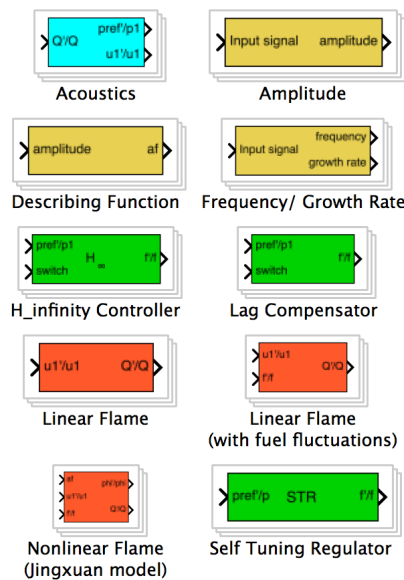


Figure 2: OSCILOS_SIM block library

The OSCILOS_SIM library

The OSCILOS_SIM library contains a collection of blocks which can be used to simulate the above combustor. These are shown in Fig. 2 and include: (i) an acoustic model; (ii) linear and nonlinear flame models; (iii) several blocks for analysing pressure oscillations, growth rate and frequency and (iv) three types of controller, including robust and adaptive controllers. The necessary parameters, including the fixed time-step of the solver, dt , are prescribed in the Matlab file *parameters.m*, which must be selected as the model workspace in the simulink model.

Two test simulations are provided to help get started; OSCILOS_Sim_lin (linear flame model) and OSCILOS_Sim_nonlin (nonlinear flame model). The latter includes adaptive feedback control. A step-by-step guide to getting these simulations working is now provided:

- Open Matlab (version 2014b or later) normally. If you are doing this through the command line, don't use 'nodesktop' or 'nojvm' options. Ensure you are working in the OSCILOS_SIM directory.
- In the Matlab command window, type 'simulink' to open Simulink. In the Simulink Library Browser window, check that OSCILOS_SIM is listed in the libraries on the left. If it isn't, you will need to add it by opening it, unlocking it, typing a command of the form 'set_param (gcs , 'EnableLBRepository' , 'on')' and possibly refreshing the Library Browser.
- Open one of the test simulation files, 'OSCILOS_Sim_lin.slx' or 'OSCILOS_Sim_nonlin.slx' e.g. by clicking on the yellow 'open file' option in the Simulink Library Browser window,.
- Before running the simulation, click the down arrow to the right of the blue/white square 'Model explorer' icon, and choose 'Model workspace'. On the left hand side of the 'Model Explorer' window, check that the 'Model Workspace' filename points to the file parameters.m in the current directory. If it doesn't, browse and choose this file so it does. Click 'Reinitialize from source' and choose the acoustic boundary conditions at the upstream and downstream tube ends by clicking in the popup windows that appear (we suggest 'Open' and 'Open' initially).
- Click the green arrow button in the simulink file window to start the simulation. When it is complete, double click on pref'/p1, frequency or growth rate blocks within the block diagram to view the time history of these. Click the 'expand' icon (four outward arrows) to see graphically.
- Play around with blocks and controllers to build your own model!