Tutorial 2: Tools from Nonlinear Dynamics

R. I. Sujith
Aerospace Engineering Department, Indian Institute of Technology Madras, Chennai 600036, India

Contents

1. Basics of Signal Processing and Types of Signals 1
2. Bifurcation Theory 2
   2.1. Linear stability analysis of a one-dimensional system: 2
   2.2. Linear stability analysis of a two-dimensional system: 3
3. Phase-Space Reconstruction: 7
   3.1. Basics of phase space reconstruction 8
4. Poincaré Section (First Return Map): 17
5. Recurrence Plot Analysis: 19
6. Complex Networks (Visibility Algorithm): 23
7. Multifractal Detrended Fluctuation Analysis (MFDFA): 27

1. Basics of Signal Processing and Types of Signals

During signal processing of real-valued signals, we encounter two types of signals: analog and digital (1). Analog signals are continuous in time. These signals are mostly obtained from analytical solutions of the governing equations of the system or sometimes from analog computer simulations or from experiments. On the other hand, digital signals are non-continuous or discrete in time. Digital signals are obtained from a discrete but even sampling of the analog signals. These signals are commonly obtained from digital computer simulations or from experiments. Real signals are analog in nature. In order to process these signals using computers, we have to convert continuous analog signals into digital signals by using analog-to-digital (A/D) converters (2; 3; 4).

A brief list of the steps usually followed in the analog to digital conversion of the signals:

1. Acquire real-valued analog signals in the form of a voltage signal using a transducer.
2. Choose the range of the voltage ($V_{max} - V_{min}$) such that all the variations of the signal (up and down) are covered in that range.
3. Sample the signal into equal intervals of time. This process converts a continuous time signal having infinitely many values into a signal of finite discrete values. Sampling theorem (5) must be followed prior to the sampling of a signal, which says that the sampling frequency should be at least twice the highest frequency of interest. Sampling is always done by using a Sample-and-Hold (S/H) circuit located at the input of the A/D system. The output signal of the S/H has a staircase-like structure.
4. A numerical value is assigned to the voltage level obtained at the output of the S/H through a quantization process. Quantization is the process of converting a continuous voltage signal into discrete numbers of voltage levels. The number of possible quantization levels is computed in the powers of 2 i.e., $N = 2^n$, where n = number of bits in the A/D converter.
5. Finally, the quantized signal is converted into a digital representation using a process of encoding. Here, to each quantization level, a unique number is given. For example, in the case of the 3-bit converter, the binary encoding is performed as 000, 001, 010, etc.
6. This digital signal is then sent to a receiver, where it is decoded and the analog signal is obtained.
7. The resolution of n-bit A/D converter is obtained as $(V_{max} - V_{min})/N$, where the analog voltage range is over the full-scale. The resolution can be increased by increasing the number of bits of A/D converter. For example, for a 16-bit A/D converter having an analog voltage range of 0 to 10 V, the resolution of the converter is $(10 - 0)/2^{16} = 10/65536 = 0.153$ mV.

In general, two types of systems are present in nature (1): 1) deterministic and 2) random.

Signals obtained from deterministic systems are classified as periodic and non-periodic. Non-periodic signals include quasiperiodic and chaotic signals. On the other hand, signals obtained from a random system are called random signals.

Signals of deterministic or random systems are further classified into two types: stationary or nonstationary (1). For stationary signals, the statistical quantities such as mean, variance or autocorrelation do not change over time (e.g., white noise, sinusoidal). In contrast, for non-stationary signals, these quantities change over time

(chirp, real-time data of EGC, finance, etc.). The behavior of non-stationary signals can be recognized as trends, cycles, random walks, or a combination of these behavior.

The characterization of the time history of a signal is important to understand whether its asymptotic behavior or final motion of the system reaches a steady state or not (6; 7; 8). Steady state here means the evolution of the signal saturates to a constant non-zero amplitude value (limit cycle) or to a zero value (fixed point). An understanding of steady state motion is important to remove the transient behavior of the signal from the analysis. Such analysis is difficult for aperiodic signals.

The different types of signals observed in practical systems can be identified from their specific features in the power spectrum (6; 8). Such identification of the signals can be performed in the following manner. If the oscillatory dissipative system is subjected to a single frequency periodic signal, and if the obtained response spectrum has

1. Only a single peak at the excitation frequency, then the system has linear periodic motion.
2. Otherwise, if the response spectrum has multiple peaks at sub or higher harmonics other than the excitation frequency, then the motion is non-linearly periodic.
3. If the response power spectrum contains two frequencies which are incommensurate, then the motion is quasiperiodic.
4. Both periodic and quasiperiodic motions have discrete peaks in the power spectrum.
5. If the spectrum is continuous and broadband, the motion could possibly be chaotic or transient or random.
6. In the case where the ratio between the two frequencies of a power spectra is a rational number then the motion is called mode-locked.

2. Bifurcation Theory

The study of qualitative changes that happen in the structural (topological) properties of a dynamical system due to the variation of a system parameter was introduced by Henri Poincaré (9). Such a mathematical study of understanding the qualitative changes in the system dynamics due to the smooth variation of a control parameter is commonly referred to as bifurcation theory. Bifurcation is a phenomenon that occurs when a small change in the system parameter results in a change in qualitative properties of the system dynamics (10). The position of the control parameter at which such changes occur in the system dynamics is called a bifurcation point. The qualitative changes that happen during a bifurcation may include sudden appearance, disappearance or change in the stability of equilibrium points, periodic or quasi-periodic orbits, or complex dynamics such as a strange attractor (11). Such changes in the system dynamics can be effectively represented by using a bifurcation plot. In the bifurcation plot, one of the statistical parameters (such as signal peaks, sample mean, sample median, etc.) of the dynamical variable is plotted against changes in the system parameter.

Bifurcations in the system dynamics are mainly classified into two types viz., local and global bifurcations. For a local bifurcation, the stability of the system dynamics is analyzed in a small portion of the phase space, which is near to the equilibrium (or fixed) points. The sign (positive or negative) or the value (real or complex) of the eigenvalues obtained from a linear stability analysis of such equilibrium points decide the type of the local bifurcations. These bifurcations include saddle-node, pitchfork, transcritical, period-doubling, Hopf, and NeimarkSacker bifurcations (12). On the other hand, during global bifurcations, the changes that happen in the phase space trajectory are not limited to the neighborhood of the equilibrium points, whereas, it is distributed over a wide portion of the phase space. Such bifurcations mainly arise due to the collision of the large invariant sets like periodic orbits with each other or with the equilibrium points (10). Examples of theses bifurcations include homoclinic, heteroclinic, infinite period doubling, etc.

As explained in the previous paragraphs, the type of bifurcations exhibited by the system is commonly dependent on the stability of the fixed points observed in the dynamical system. One way to understand the stability of such fixed points is using a linear stability analysis (10). Here, the solutions of the governing differential equations representing a given dynamical system are approximated about the fixed points. By using a weak perturbation analysis, the divergence or convergence of these perturbations is detected. A procedure of performing linear stability analysis for the first order differential equations of a single and double variable dynamical system is explained in the following sections. The description is based on the reference textbooks of Strogatz (10), Hilborn (6), Orlik (13).

2.1. Linear stability analysis of a one-dimensional system:

Let us consider a first order differential equation of a single variable dynamical system as $\dot{x} = f(x)$. The roots of this equation (i.e., when $f(x) = 0$) are called fixed points or equilibrium points. Let x^* be a fixed point (i.e., $f(x^*) = 0$) and $\eta(t)$ be a small perturbation to x^* such that $\eta(t) = x(t) - x^*$. Therefore, the stability of this

fixed point is decided by the time evolution (growth or decay) of the perturbation.

The time derivative of the perturbation calculated as $\frac{d\eta}{dt} = \dot{\eta} = \frac{d(x(t)-x^*)}{dt} = \dot{x}$. Here, $\frac{dx^*}{dt} = 0$, as x^* is a constant. Therefore, $\dot{\eta} = \dot{x} = f(x) = f(\eta + x^*)$.

By Taylor's expansion of the right hand side of the above equation, we get, $f(\eta+x^*) = f(x^*)+\eta f^{'}(x^*)+O(\eta^2)$, where, $O(\eta^2)$ denotes the higher order terms of η.

Since $f(x^*) = 0$, we get, $\dot{\eta} = \eta f^{'}(x^*) + O(\eta^2)$.

If $f^{'}(x^*) \neq 0$ and on neglecting $O(\eta^2)$, we can write the linearized equation of the given dynamical system about the fixed point as $\eta \approx \eta f^{'}(x^*)$.

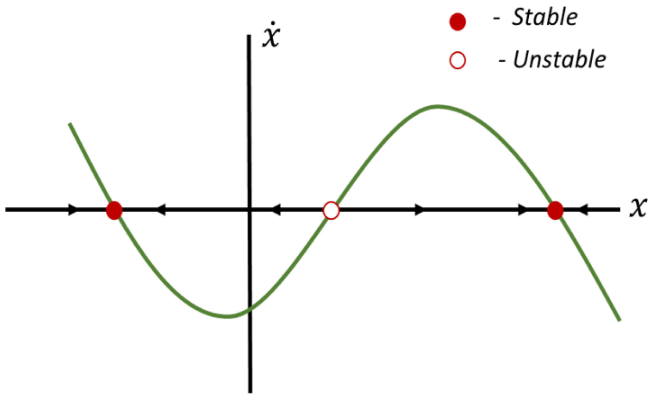

Figure 1

The linear stability analysis of the fixed points of a first order dynamical system. The fixed points could be stable (shown by a filled circle) or unstable (shown by an empty circle). The stable fixed point is an attractor, where all the neighborhood trajectories move towards this point. The unstable fixed point is a repeller, where all the trajectories in the neighborhood of this point move away.

The stability of this equation depends on the value of $f^{'}(x^*)$. If $f^{'}(x^*) > 0$, the perturbations (η) grow exponentially with time and the solution of the system is referred to as unstable, whereas, if $f^{'}(x^*) < 0$, the perturbations (η) decay with time and the solution of the system is referred to as stable. When $f^{'}(x^*) = 0$, we need to include higher order terms to decide the stability of the fixed points. This behavior of stability of the fixed points can be easily seen in Fig. 1. Whenever the slope of the x^* versus x plot is negative, the fixed point is stable and all the neighboring trajectories to this point will attract to each other. On the contrary, when the slope of x^* versus x plot is positive, the fixed point is unstable and hence all the neighboring trajectories move away from that fixed point.

2.2. Linear stability analysis of a two-dimensional system:

Let us consider the equations of a two-dimensional linear dynamical system given by, $\dot{x} = f(x, y) = ax + by$, and $\dot{y} = g(x, y) = cx + dy$, where a, b, c, d are the parameters. These equations can be written in the matrix form as follows,

$\dot{X} = AX$, where, $\dot{X} = \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix}$, $X = \begin{pmatrix} x \\ y \end{pmatrix}$, and $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$.

Thus, the solution of $\dot{X} = AX$ can be visualized in terms of the movement of the phase space trajectory in the (x, y)-plane.

Using the same linear stability analysis, as explained in the previous section for a single variable, a Jacobian matrix for two variable system can be obtained as,

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} f_x(x^*, y^*) & f_y(x^*, y^*) \\ g_x(x^*, y^*) & g_y(x^*, y^*) \end{pmatrix},$$

where, x^* and y^* are the fixed points, and f_x, f_y, g_x, g_y are the first order derivatives of f and g with respect to x and y respectively.

Let, $\tau = Tr[A] = a + d$ and $\Delta = det(A) = ad - cb$, then the eigenvalues of the matrix A can be obtained from the solution of a characteristic equation $det(A - \lambda I) = 0$ as follows, where, λ is an eigenvalue and I is an identity matrix.

On solving $\begin{vmatrix} a-\lambda & b \\ c & d-\lambda \end{vmatrix} = 0$, we get, $\lambda^2 - \tau\lambda + \Delta = 0$. Further, the general roots of the characteristic equation are obtained as, $\lambda_1 = \left(\tau + \sqrt{(\tau^2 - 4\Delta)}\right)/2$, and $\lambda_2 = \left(\tau - \sqrt{(\tau^2 - 4\Delta)}\right)/2$.

Thus, the linear stability or the behavior of the two-dimensional dynamical system in the phase space is

decided by the sign of real parts of both λ_1 and λ_2. The different types of possible fixed points observed in a two-dimensional dynamical system are listed below and also shown in figure 2.

- If both $Real(\lambda_1)$ and $Real(\lambda_2) < 0$, i.e., the eigenvalues are real and negative (the upper left quadrant of the (τ, Δ) - plane), the fixed points are stable.
- If $Real(\lambda_1) < 0$ and $Real(\lambda_2) > 0$, i.e., the eigenvalues are real and are of opposite signs (the entire lower half of the (τ, Δ) plane), the fixed point is called a saddle (unstable point).
- If both $Real(\lambda_1)$ and $Real(\lambda_2) > 0$, i.e., the eigenvalues are real and positive (the upper right quadrant of the (τ, Δ) - plane), the fixed points are unstable.
- The parabola $\tau^2 - 4\Delta = 0$, separates spirals from nodes. The star and degenerate nodes (stable/unstable) are located along the parabola line.
- If both λ_1 and λ_2 are complex conjugates of each other (i.e., $\lambda_{1,2} = \alpha \pm i\omega$, where $\alpha = \tau/2$ and $\omega = \frac{\sqrt{(\tau^2-4\Delta)}}{2}$, the fixed point is called a spiral (a plane enclosed by the parabola). Here, if $Real(\lambda_{1,2}) < 0$, it is called a stable spiral (decaying solution), and if $Real(\lambda_{1,2}) > 0$, it is called an unstable spiral (growing solution). If the eigenvalues are purely imaginary ($\tau = 0$), all solutions are periodic, then the corresponding fixed point is called a center (the points along the positive Δ - axis).
- If $\lambda_1 = \lambda_2 \neq 0$, all trajectories passing through the fixed points are straight lines and the corresponding fixed point is called a star node.
- If $\lambda_1 = \lambda_2 = 0$, the whole phase plane is filled with the fixed points.
- If $\lambda_1 \neq 0$ and $\lambda_2 = 0$ i.e., only one eigendirection exists, the fixed point is called a degenerate node.
- In the case where all the eigenvalues of a dynamical system have a non-zero real part, the corresponding system is referred as hyperbolic.

As we know, during a bifurcation process, a small change in the control parameter results in a change in the qualitative behavior of the system dynamics. The different kinds of such qualitative behavior (or bifurcations) exhibited by the dynamical system are further classified into four main types as saddle-node, transcritical, pitch-fork, and Hopf bifurcation (10). A brief description of these types of bifurcations is provided in the following paragraphs.

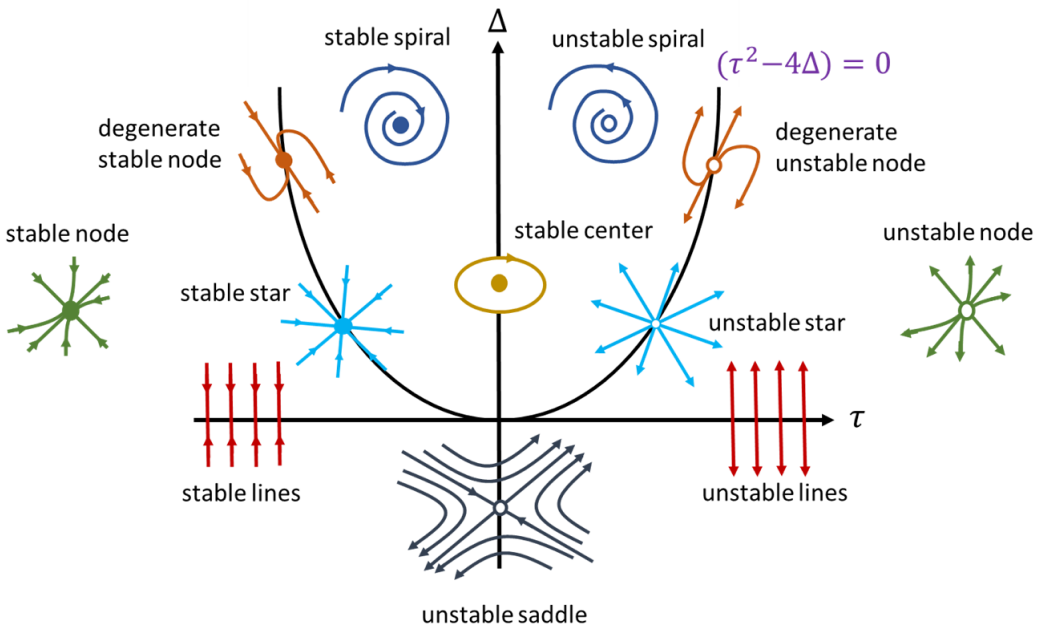

Figure 2

A schematic of the possible fixed points observed in a two-dimensional dynamical system. The classification of fixed points is based on the sign (positive or negative) or value (real or complex) of the solutions obtained from linear differential equations of the system. These fixed points are mainly classified into four types as node, spiral, center, and saddle. The nodes and spirals are either asymptotically stable or unstable. A center is stable but not asymptotically stable, whereas, a saddle is an unstable point.

2.2.1. Saddle-node (or fold or tangent) bifurcation. In this bifurcation, fixed points of the dynamical system are created and destroyed as the control parameter is changed. Here, two fixed points initially move close to each other, due to a change in the system parameter, merge together and eventually vanish (see Fig. 3). For example,

consider a first order dynamical system, $\dot{x} = r + x^2$, where r is the control parameter. The fixed points of the given dynamical system are $x^* = 0, \pm\sqrt{(-r)}$. Here, when $r > 0$, the equation has no possible solution. When r is negative, there are two fixed points as $\pm\sqrt{-r}$. Out of these fixed points, $-\sqrt{-r}$ is a stable and $+\sqrt{-r}$ is an unstable fixed point. When $r = 0$, both fixed points merge together, and when $r > 0$, the system has no possible fixed points.

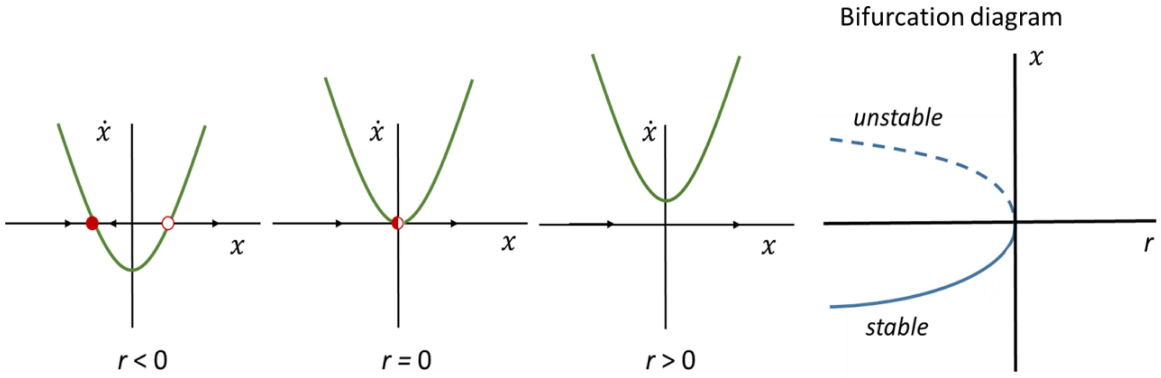

Figure 3

A schematic of the bifurcation process of the saddle-node type shown by a dynamical system due to change in the control parameter (r). Two individual fixed points eventually come closer, merge and vanish as the control parameter is changed gradually from negative to positive value respectively. This results in two separate stable and unstable branches in the bifurcation point when r is less than zero and, no stability branch when r is greater than zero.

2.2.2. Transcritical bifurcation. In this case, all the fixed points of a system never get destroyed during the variation of the control parameter. With the change in control parameter, these fixed points just exchange their stability (see Fig. 4). For instance, a dynamical system is given by the equation, $\dot{x} = rx - x^2$. Here, the fixed points are $x^* = 0$ and r, where, $x^* = 0$ always remains a fixed point for all values of r (positive or negative). When $r < 0$, $x^* = 0$ is a stable fixed point and $x^* = r$ is an unstable fixed point. On the other hand, for $r > 0$, these fixed points exchange their stabilities such that $x^* = 0$ becomes an unstable fixed point and $x^* = r$ becomes a stable fixed point. At $r = 0$, there is only one fixed point at $x^* = 0$.

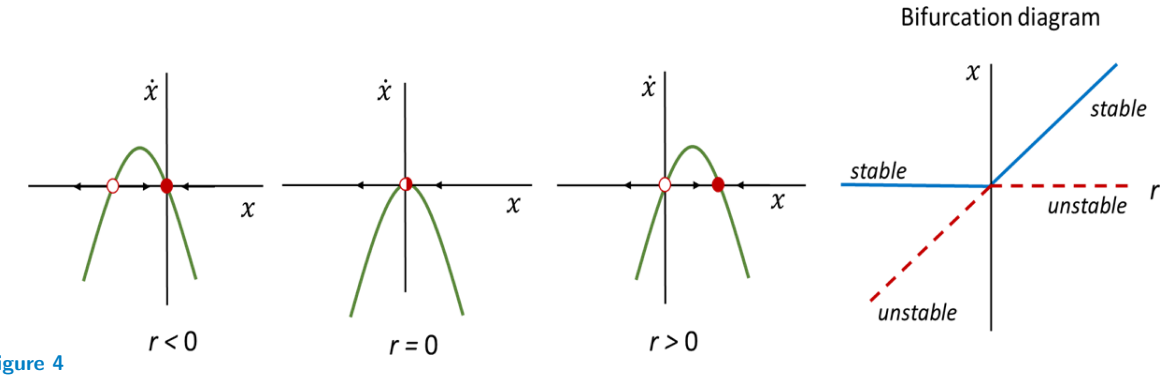

Figure 4

A schematic of the bifurcation process of a transcritical bifurcation exhibited by a system due to change in the control parameter. As the control parameter is changed from negative to a positive value, the fixed points exchange their stabilities after the zero crossing by the control parameter, i.e., the stable fixed point becomes unstable and vice versa. In the bifurcation plot, this exchange of stability is clearly visible, where the stable branch becomes unstable and the unstable branch becomes stable as we change the control parameter from negative to positive value.

2.2.3. Pitchfork bifurcation. This type of bifurcation is commonly observed in the systems wherein the symmetry in their underlying dynamics is retained under the transformation, $x \to -x$. Let us consider a system whose equation is $\dot{x} = rx - x^3$, where fixed points are $x^* = 0, \pm\sqrt{r}$. When $r < 0, x^* = 0$ is the only fixed point and, it is stable. For $r = 0, x^* = 0$ is still a fixed point and it is weakly stable, which means the solution does not decay exponentially fast, instead the rate of decay is much slower. This phenomenon is referred as critical slowing down in the physics literature (10). For $r > 0$, two other fixed points, $x^* = \pm\sqrt{r}$ appear, in addition to $x^* = 0$, where $\pm\sqrt{r}$ are stable fixed points and 0 is the unstable fixed point. Thus, the appearance of two fixed points symmetrically about the origin is a property of the pitchfork bifurcation. This type of bifurcation is

further classified into two types as supercritical pitchfork bifurcation and subcritical pitchfork bifurcation. The bifurcation diagram of these two types of bifurcations is given in Fig. 5.

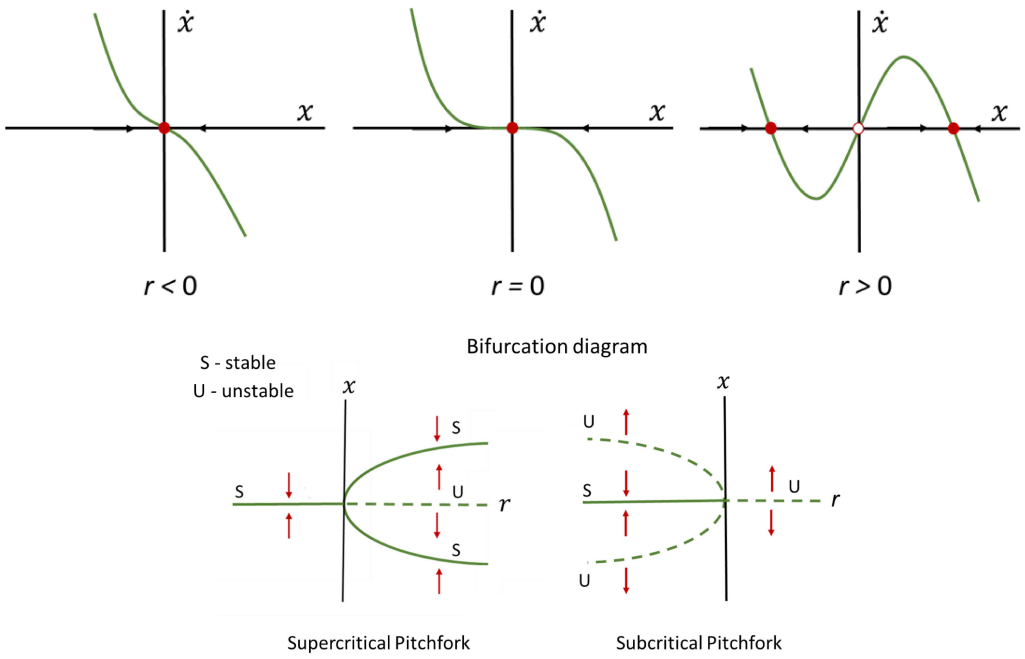

Figure 5

A schematic of the bifurcation process of a transcritical bifurcation exhibited by a system due to change in the control parameter. As the control parameter is changed from negative to a positive value, the fixed points exchanges their stabilities after the crossing of a zero value by the control parameter, i.e., the stable fixed point becomes unstable and vice versa. In the bifurcation plot, this exchange of stability is clearly visible, where the stable branch becomes unstable and the unstable branch becomes stable as we change the control parameter from negative to positive value respectively.

2.2.4. Hopf bifurcation (or Poincaré-Andronov-Hopf bifurcation). This type of bifurcation occurs when the complex conjugate eigenvalues of a dynamical system cross an imaginary axis as the control parameter is varied. This leads to a generation of the self-sustained limit cycle (periodic) oscillations from a stable fixed point in the system. When the real part of the eigenvalues is negative, the fixed point acts as a stable focus. Whereas, when the real part of eigenvalues becomes positive due to the crossing of imaginary axis, the fixed point becomes an unstable focus, where all the trajectories originating from the fixed point are attracted toward the stable limit cycle.

Hopf bifurcation is further classified into two types as supercritical or subcritical bifurcations. This classification is based on whether the stable limit cycle is surrounded by unstable fixed point (supercritical Hopf) or the stable fixed point is surrounded by the unstable limit cycle (subcritical Hopf).

The classification of supercritical and subcritical bifurcation is based on a few general distinguishing properties illustrated by both of these bifurcations (10) as explained below. In supercritical bifurcation, the size of the generated limit cycle during bifurcation continuously increases from zero with an increase in the control parameter. On the other hand, during subcritical bifurcation, there is a sudden generation of large amplitude limit cycle oscillations in the system. Because of this property, the occurrence of subcritical bifurcations is dangerous in practical applications. Supercritical bifurcations do not possess hysteresis, whereas, subcritical bifurcations always have a hysteresis (or bistable) region.

The normal form of Hopf bifurcation in cartesian coordinate is given by $z = (\mu+i)z \pm z|z|^2$, where $z = x_1 + ix_2$ and is a parameter. In polar coordinates, this normal form can be written as $r = \mu r \pm r^3$, $\dot{\theta} = 1$, where $z = re^{i\phi}$. In these normal forms, the '+' sign corresponds to subcritical Hopf bifurcation and the '−' sign corresponds to supercritical Hopf bifurcation. For example, in the case of supercritical Hopf bifurcation, the normal form in polar coordinate is given as $r = \mu r - r^3$, $\dot{\theta} = 1$. There are three fixed points for the first equation: $r = 0$ and $r = \pm\sqrt{\mu}$. Among these fixed points, $r = 0$ is always a fixed point for all values of μ. For $\mu < 0$, the fixed point is linearly stable; for $\mu = 0$, the fixed point is nonlinearly stable (this means that the solution does not converge to zero exponentially); for $\mu > 0$, the fixed point is linearly unstable. In the case of $r = \pm\sqrt{\mu}$, the fixed points are stable only for $\mu > 0$ and no solution exists for $\mu < 0$. The second equation ($\dot{\theta} = 1$) represents the rotation at a constant speed. The superposition of these equations can help in demonstrating a supercritical Hopf bifurcation plot for a two-dimensional system as shown in figure 6(b). The origin of the system will always be a fixed point.

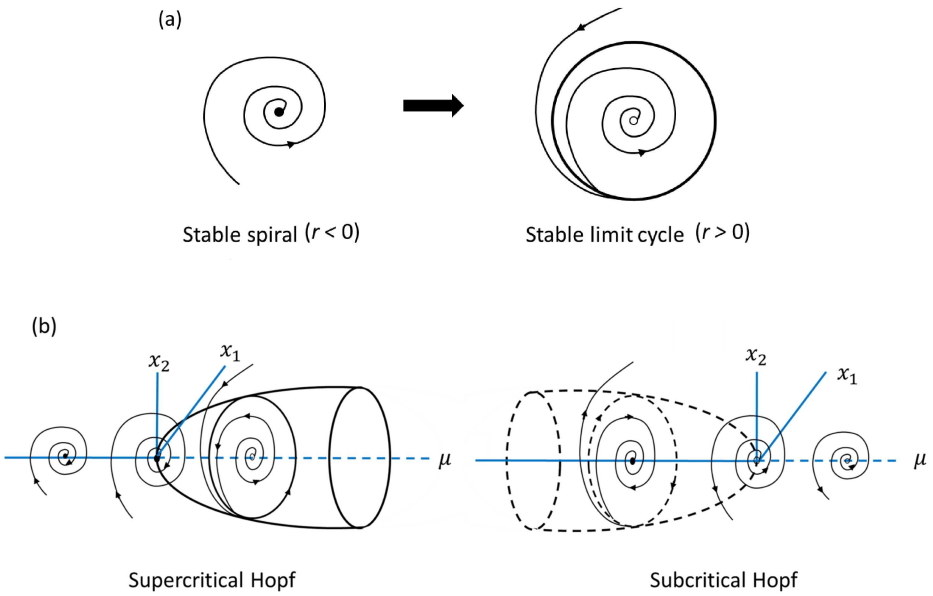

Figure 6

(a) Schematic of the process of Hopf bifurcation due to change in the control parameter. The change in control parameter, from negative to positive value, results in the transition of a stable fixed point to an unstable fixed point along with the generation of a stable limit cycle attractor in the system. (b) The Hopf bifurcation is classified into supercritical and subcritical Hopf bifurcations. The continuous lines in the figure correspond to stable dynamics and the dotted lines correspond to unstable dynamics of the attractor. During supercritical Hopf bifurcation, single stable branch bifurcates into a stable limit cycle and an unstable branch. The reverse happens during the subcritical Hopf bifurcation.

For $\mu \leq 0$, the fixed point is the stable focus where all the trajectories spiral towards the fixed point, and for $\mu > 0$, the fixed point is always an unstable focus where all trajectories spiral out from the fixed point and attract towards a stable limit cycle attractor of radius $r = \sqrt{\mu}$. A similar explanation can be obtained for the subcritical Hopf bifurcation.

The onset of thermoacoustic instability in combustion systems is viewed as a Hopf bifurcation in general (14; 15; 16). Depending on the operating condition, Lieuwen (17) observed both subcritical as well as supercritical Hopf bifurcation in the bluff-body-stabilized turbulent combustor. Etikyala and Sujith (18) experimentally studied the change of criticality in a horizontal Rijke tube and proposed a model that captures this behavior. Lipika et al. (19), in a ducted laminar premixed burner, reported the presence of various types of secondary bifurcations such as limit cycle, period-doubling, quasiperiodicity, period-k and chaotic oscillations happening through a subcritical Hopf bifurcation of the fixed point.

3. Phase-Space Reconstruction:

Dynamical systems are systems that evolve with time. The dynamics of such systems at every instant of time can be appropriately determined from the available knowledge of fixed number of independent variables (also known as state variables) that characterize the system. The time evolution of these variables can be expressed in the form of a set of linear differential equations given by $\dot{X} = \Phi(X)$, where the dot represents the time derivative and the function Φ comprises the overall form of dynamical evolution of that variable. For a given set of initial conditions and the functional form, the future dynamical states of the system variable can be correctly determined from the above equation. The path traced by the sequence of points (also called states) obtained from the solution of this equation is called the trajectory of the system dynamics. The evolution of this trajectory is often represented in a plot known as a phase portrait. A phase portrait is a plot between the number of independent variables that are required to signify the state of a system correctly. Phase space includes all possible states of the system dynamics. For instance, in a mechanical system, a system with two degrees of freedom is shown in the phase space by a plot between a state variable (x - displacement) and its time derivative ($\dot{x}$ - momentum). The evolution of the phase space trajectory could be unbounded (extend to infinity with an increase in time) or bounded (trapped in a region for all the initial conditions). The latter case of the dynamical evolution is known as the basin of attraction (or attractor) of the trajectory.

A list of various types of dynamical motions exhibited by the system and the types of attractors present in the phase space is shown below:

1. Fixed point: A fixed point corresponds to a point in the phase space. In the experimental situation, a fixed

point corresponds to a static equilibrium position of the system (8). The minimum embedding dimension required for visualizing the fixed point is 1.

2. Periodic oscillations (or limit cycle attractor): The dynamics of such oscillations is often identified in terms of a closed loop in a phase space. Different features of the periodic motions can be seen in the phase space. If the state variables of the system are periodic and contain oscillations of the same frequency, then their plot in two-dimensions would look like either a straight diagonal line, a circle or an ellipse, depending on the value of the time delay (8). The distortions in the phase plot are the artifact of the presence of harmonics in the signals. The minimum embedding dimension required to visualize periodic motion is 2.

3. Quasiperiodic oscillations (or torus attractor): These motions correspond to the appearance of toroidal structure in the phase space. Such motion primarily arises due to the competition between two or more independent frequencies that are incommensurate with each other (6). In the case of externally forced systems, the competing frequencies are the frequency of forcing and that of the natural oscillations in the system. Whereas, in the situation where oscillations are spontaneously generated in the system, the competing frequencies could be different modes or frequencies of the same system. The minimum embedding dimension required to observe an attractor of the quasiperiodic motion is 3 or higher.

4. Chaotic oscillations (or strange attractor): The attractor of such oscillations does not have any definite geometric structure in the phase space. Such oscillations are often aperiodic and, mainly arise in the system dynamics due to competition between three or more frequencies. The attractor of such motions possess extremely complicated dynamics (hence, sometimes referred as a strange attractor (20)) because of the presence of many crossing and loops in the phase trajectory. Chaotic motions are sensitive to initial conditions; a small change in the initial condition would lead to an exponential divergence of the neighboring trajectories in the phase space. The minimum embedding dimension necessary to visualize chaotic motions is 3 or higher.

5. Intermittency: The motion of such dynamics often switches intermittently back and forth between two qualitatively different behavior, for instance, regular and chaotic. Such switching of the system dynamics happens at a constant value of the control parameter even in the absence of external noise (6). The switching of system dynamics appears seemingly random during intermittency. Intermittency has been mainly categorized into three types: type-I, type-II, and type-III (21). These types of intermittency correspond to specific kind of bifurcations as, for type-I intermittency - it is a saddle-node bifurcation, for type-II intermittency it is a subcritical Hopf bifurcation, and for type-III intermittency - it is a period doubling bifurcation.

3.1. Basics of phase space reconstruction

Real systems are more complex and require an infinite number of dimensions to describe the state of a given dynamical system completely (22). In practice, the number of dynamical variables available from a given system is restricted to a few numbers, and in the limiting case, it is one. In order to visualize and subsequently characterize asymptotic motions of such variables in a higher dimension, a method known as phase space reconstruction (23) has been popularly used in the literature of dynamical systems theory (24; 8). For example, we have a time sampled measurement of a single dynamical variable acquired from the system as $x(t) = x(t_0 + n\tau_s)$, where t_0 is the initial condition and τ_s is the sampling time. In such a situation, it is possible to construct a higher dimensional phase space, from the given knowledge of a single univariate observation, using a theorem of time delay embedding proposed by Takens (1981). Here, embedding is a process of one-to-one mapping of points on the attractor from its original dimensional space to the attractor in the reconstructed space (25). The embedding should be such that it preserves all the topological information as well as the dynamical invariants such as Lyapunov exponent and correlation dimension of the original system in its reconstructed space. In principle, the dimension of the attractor should be sufficiently high so that the trajectories of the reconstructed space do not show any overlap.

According to Takens delay embedding theorem (26), if we have a single observable time series $[x_1, x_2, x_3, \cdots, x_N]$ measured at a particular value of the control parameter, we can embed the signal into a higher-dimensional phase space by choosing an appropriate value of time delay (τ) and embedding dimension (d). With this, the constructed delay-coordinate vectors can be expressed as, $Z(i) = [x(i), x(t+\tau), x(i+2\tau), x(i+3\tau), \cdots, x(i+(d-1)\tau)]$; where, $i = 1, 2, 3, \cdots, N-(d-1)\tau$. The plot between the elements of these vectors shows the evolution of the system dynamics in the d-dimensional phase space.

```
function [Y]=ARFM_delay_vec(x,tao,d,N1)
```

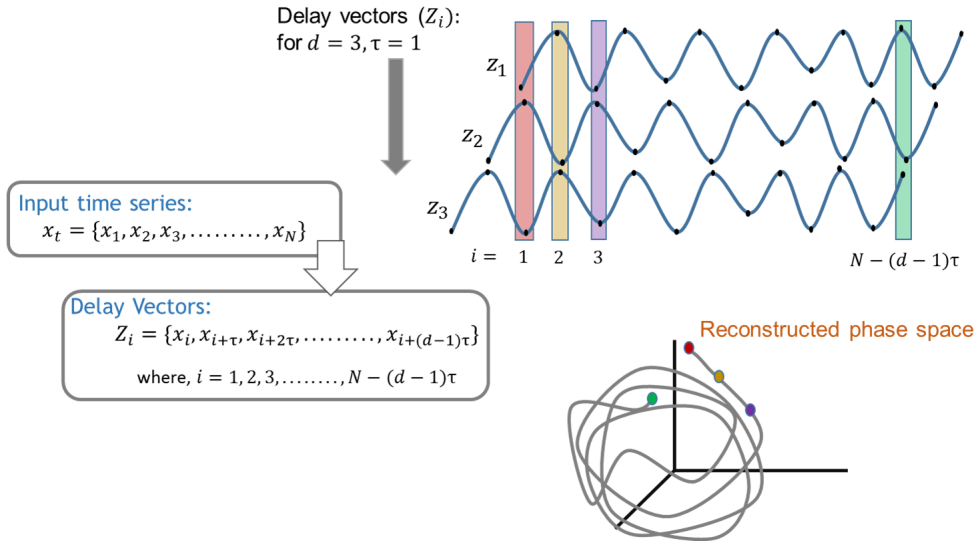

Figure 7

The schematic of the process of phase space reconstruction of a univariate time series using the time-delay embedding theorem proposed by Takens (26).

```
% This code constructs the delay vectors from a give scalar time series using
    appropriate value of embedding
% dimension and the delay time. It is based on the theorem of time delay embedding
    proposed by Takens (1981)

% ------------- Inputs -----------------
%  x:      A scalar time series
%  tao:      The delay time (calculated from AMI)
%  d:      The embedding dimension (calculated either from FNN method or Cao's method)
%  N1:     Number of delayed vectors (N1=length(x)-(d1-)*tao)
% ------------- Outputs ---------------
%  Y: A matrix of dimension [N1,d], where N is the number of delayed vectors calculated
    from the signal
% and d is the dimension in which the signal is embedded
%%----------------------------------
% Initializing a delay vector matrix, where 'N1' is the number of delay vectors and 'd'
    is the embedding dimension
Y=zeros(N1,d);
for i=1:d
    Y(:,d-i+1) = x(1+(i-1)*tao:N1+(i-1)*tao);  % Delay vectors are creased by lagging
    the signal with a given delay time
end
```

Listing 1 Function to construct delay vectors from a given scalar time series; `ARFM_delay_vec.m`

3.1.1. Selection of optimum time lag (τ) and minimum embedding dimension (d). The selection of proper reconstruction parameters such as τ and d is not straightforward. The choice of some pairs of these parameters will have much clearer results than others (8). There are a few popular methods available for choosing the optimum values of these parameters and have been listed below.

a. Choosing the optimum time lag (τ):

Theoretically, for infinite noise-free data, any arbitrary value of the successive measurement can be chosen as the optimum time delay, according to Takens embedding theorem (26). However, in reality, due to practical limitations such as finite data length, finite precision and the presence of noise, the choice of time delay is not straightforward (28). As every acquired signal contains new information in their successive measurements of time, the choice of time delay should be such that it will produce independent delayed coordinates in their reconstructed phase space (29). If the delay is small (compared to the time scale of the system), it will produce highly correlated delayed vectors. If the delay is too large, all the delayed vectors will become completely uncorrelated and the reconstructed phase space will not represent the true dynamics of the system (24). The delay should not be close to the time period (inverse of the frequency) of the signal (8). If the delay is equal to the time period of the signal, as used in the construction of a Poincaré map, the corresponding periodic component will not be represented properly in the reconstructed phase space. Thus, the choice of the delay is tricky; its value should be sufficiently large so that the delayed vectors are independent, but not so large that they would become completely

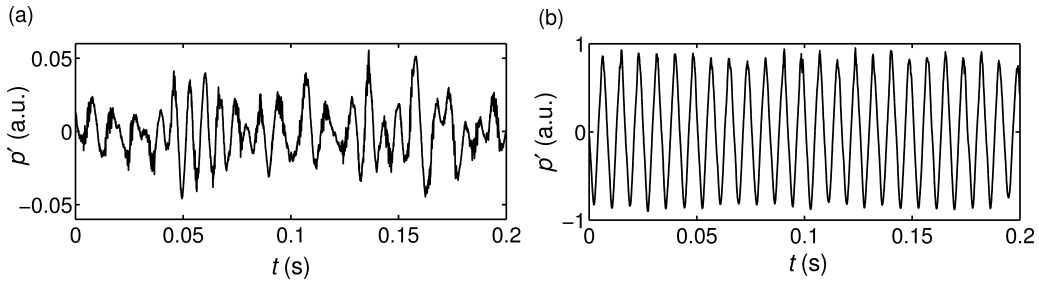

Figure 8

(a), (b) The time series of the acoustic pressure obtained from a combustor with turbulent flow during the state of combustion noise and thermoacoustic instability, respectively. The state of combustion noise corresponds to low amplitude aperiodic oscillations, whereas, that of thermoacoustic instability corresponds to high amplitude periodic oscillations in the acoustic pressure of the system. (The data is published in ref. (27)

independent. In reality, we have finite equally sampled data with a sampling frequency of $1/\tau_s$, the delay should always be expressed in terms of multiples of sampling time (τ_s).

There are two practical ways to determine the optimum time delay required for reconstruction of the phase space: 1) autocorrelation function (30) and 2) average mutual information (31).

1. *Autocorrelation function* (ACF): For a given time series data Y_i sampled at equal intervals of time τ_s, the autocorrelation function at lag k is given by

$$r_k = \frac{\sum_{i=1}^{N-k}(Y_i - \bar{Y})(Y_{i+k} - \bar{Y})}{\sum_{i=1}^{N-k}(Y_i - \bar{Y})^2} \tag{1}$$

where, $i = 1, 2, 3, \cdots, N$ and $\bar{Y}$ is a sample mean of the signal.
Therefore, the optimum time delay can be chosen as either the time instance corresponding to the first zero crossing of r_k or the first local minima of r_k. This measure of finding the time delay is based on linear theory (24; 8) and thus it detects only linear independence of the coordinates. For nonlinear systems, average mutual information is another measure which is often preferred over ACF for finding the optimum time delay, and is based on information theory (31).

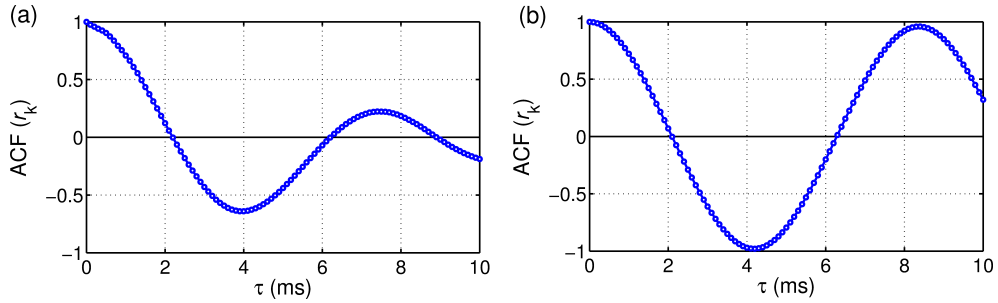

Figure 9

(a), (b) The variation of autocorrelation function (r_k) with different time lag values (τ) for the acoustic pressure signals shown in Fig. 8 obtained during combustion noise and thermoacoustic instability states respectively. The time delay corresponding to zero crossing of r_k is considered to be the optimum time delay required for the phase space reconstruction. Therefore, the value of τ for (a) is 2.2 ms and that for (b) is 2.1 ms.

2. *Average Mutual Information* (AMI): Average mutual information is a popular method used in the dynamical systems' literature to find the optimum time delay (τ) required for the reconstruction of a phase space. This concept is based on information theory. The first local minima of AMI is chosen as the optimum time delay (31). AMI is a generalized version of the autocorrelation function. It measures the extent to which $x(t+\tau)$ is related to $x(t)$ at a given τ.
AMI indicates how much information we can obtain from the measurement of a_i, chosen from set A, about the measurement of b_i, chosen from set B. If the original signal and its delayed vector are given by $[p(t_1), p(t_2), \cdots, p(t_i)]$ and $[p(t_1+\tau), p(t_2+\tau), ..., p(t_i+\tau)]$ respectively, then the AMI between two variables can be calculated as (31),

$$I(\tau) = \sum_{p(t_i), p(t_i+\tau)} P(p(t_i), p(t_i+\tau)) \log_2 \left[\frac{P(p(t_i), p(t_i+\tau))}{P(p(t_i))P(p(t_i+\tau))} \right] \tag{2}$$

where $P(p(t_i))$ and $P(p(t_i+\tau))$ are the marginal probabilities, and $P(p(t_i), p(t_i+\tau))$ is a joint probability of occurrence of $p(t_i)$ and $p(t_i+\tau)$ in the signal.

When τ is large, the measurements become completely independent, and in the limiting case as $\tau \to \infty, I(\tau) \to 0$. In the situation of monotonous decrease of AMI (i.e., AMI does not possess distinct minima), a criterion of $\frac{I(\tau)}{I(0)} = 1/e$ can be used as a rough estimation of τ (24).

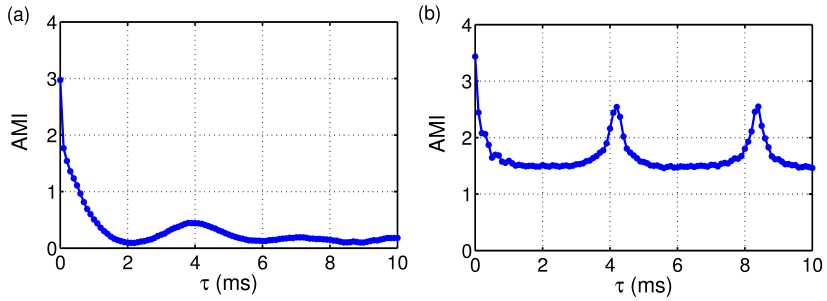

Figure 10

(a), (b) The plots representing the variation of average mutual information of the acoustic pressure signals (shown in Fig. 8), representing the states of combustion noise and thermoacoustic instability respectively, at different values of the sampling time (τ). The choice of optimum value of the time delay for (a) is 2.2 ms, and for (b) is 2.2 ms.

```
function [v, L] = ARFM_ami(x, lag)
% This code will help in finding the optimum time lag required for the
% construction of phase portrait from a discrete time series.
% _____________ Inputs ______________________
% x:        The 1D time series
% lag:      Number of lags
% _____________ Output _____________________
% v:        Average mutual information (AMI)
% _____________ Reference ___________________
% Fraser, A.M. and Swinney, H.L., 1986. Independent coordinates for
% strange attractors   from mutual information.
% Physical review A, 33(2), p.1134.
%___________________________________________
% This concept is based on information theory.
% AMI suggests how much information we can obtain from the
% measurement of a(i), chosen from set A, about the measurement
% of b(i), chosen from set B.
% The first local minima of AMI is chosen as the optimum time delay
% (Fraser & Swinney 1986).
%%-----------------------------------------------------------
y = x;      % created new vector same as given data set
x = x(:);   % transform original vector from raw to column
y = y(:);   % transform new vector from raw to column
N = length(x);  % calculate number of data points present in the signal

% Specify the value of time lag (number of samples in integer) up to
% which the mutual information is needed to estimate
L = 0:lag;
% Normalize x and y to lie between 0 and 1
x = x - min(x);
x = x*(1-eps)/max(x);
y = y - min(y);
y = y*(1-eps)/max(y);
v = zeros(size(L));     % Initialize the AMI vector

for i = 1:length(L)
    % Define the number of bins
    % Split the sample interval in log2 spaced bins
    % The logarithm of basis 2 makes the measurement of information in bits.
    bins = ceil(log2(N - L(i)));
    % Distributing data in the bin in which it is located.
    % The numbering goes from 1 to number of bins, i.e., 1, 2, 3, ..., so on
    binx = floor(x*bins) + 1;
    biny = floor(y*bins) + 1;
    % Initialize the joint probability density vector
    Pxy = zeros(bins);
    % Find joint probability Pxy for different lags
```

```
    for j = 1:N - L(i)
        k = j + L(i);
        Pxy(binx(k), biny(j)) = Pxy(binx(k), biny(j)) + 1;
    end
    % Normalize Pxy
    Pxy = Pxy/(N - L(i));
    % In order to avoid probability value tending to infinity (lag(0)), a floating
    point  number is added
    Pxy = Pxy + eps;
    % Calculating the marginal probability of x and y vectors
    % Column sum for marginal probability of x
    Px = sum(Pxy,2);
    % Row sum for marginal probability of y
    Py = sum(Pxy,1);
    % Calculating Average Mutual Information of a given time series at every time
    lag
    q = Pxy./(Px*Py);
    AMI = Pxy.*log2(q);
    v(i) = sum(AMI(:));     % [Eq. 9 in Fraser & Swinney, 1986]
    %      Shannon entropy can be calculated from following equations
    %    [Eq. 2 in Fraser & Swinney, 1986]
    %     Hx = -sum(Px.*log2(Px));
    %     Hy = -sum(Py.*log2(Py));
    %     Hxy = -sum(Pxy(:).*log2(Pxy(:)));
    %     v(i) = Hx+Hy-Hxy;
end
end
```

Listing 2 Function for calculating optimum time lag (τ) through Average Mutual Information; `ARFM_ami.m`

b. Choosing the embedding dimension (m):

The embedding dimension is an important parameter in the reconstruction of phase space. The dimension, in which, the attractor is unfolded need to be sufficiently large (24). This is important to remove false crossings and ensure the presence of true neighbors for every trajectory in the reconstructed phase space (8). It means that, if the attractor is unfolded in the d-dimensional space, the neighbors of each trajectory of the reconstructed attractor have same neighbors as that observed in the $d-1$ phase space. These neighbors are the result of their geometric nature and not the artifact of their reconstruction in a smaller dimensional space. If we know the original dimension of the system, say m (known from the m-coupled differential equations of the system), then according to Takens embedding theorem (26), the $d > 2m$ dimension is sufficient to reconstruct the observed motion of the system. However, in reality, the governing equations of the system are unknown, and due to the unavailability of data for very large durations from the experiments, it is quite difficult to find the appropriate dimension of the unfolded attractor. The common methods used to find the optimum embedding dimension required for the reconstructed phase space are explained in the following section.

1. *False Nearest Neighbors* (FNN):
   One of the popular ways to estimate the minimum embedding dimension required for the reconstruction of a phase space is based on the concept of false nearest neighbors (25). The main idea of this method is to eliminate false crossings of the phase space trajectories, which may arise due to the projection of a higher-dimensional original attractor into a lower-dimensional reconstructed phase space, and not due to their dynamics. The method of FNN measures the percentage of closeness, in terms of Euclidian distances, of neighboring points of the trajectory in a given dimensional space, and compares it with the next dimensional space. If the ratio of these distances is greater than a predefined threshold due to a change in the dimension, the neighbors of the trajectory are considered as false neighbors (25). The value of the predefined threshold should be sufficiently large so that it will allow the exponential divergence of the chaotic signal. In practice, the value of the threshold is chosen between 10 to 50 (24).
   Let the dimension of an attractor be d, the time delay be T, and the r^{th} nearest neighbor of reconstructed vector $y(n)$ is given by $y^r(n)$, then the square of the Euclidian distance between the points $y(n)$ and $y^r(n)$ is calculated as

$$R_d^2(n,r) = \sum_{k=1}^{d} [x(n+kT) - x^{(r)}(n+kT)]^2 \tag{3}$$

   where, $y(n) = [x(n), x(n+T), \cdots, x(n+(d-1)T)]$ and $n = 1, 2, \cdots, N-(d-1)T$. In the $d+1$ dimensional space, the distance between these points becomes

$$R_{d+1}^2(n,r) = \sum_{k=1}^{d+1} [x(n+kT) - x^{(r)}(n+kT)]^2 \tag{4}$$

or

$$R_{d+1}^2(n,r) = R_d^2(n,r) + [x(n+kT) - x^{(r)}(n+kT)]^2 \tag{5}$$

If the distance between $y(n)$ and $y^r(n)$ increases as we change the dimension from d to $d+1$, it implies that the given neighbor is a false neighbor in the d-dimension space.
The criterion to detect false neighbor:

$$\frac{[R_{d+1}^2(n,r) - R_d^2(n,r)]^{1/2}}{R_d^2(n,r)} = \frac{|x(n+dT) - x^{(r)}(n+dT)|}{R_d(n,r)} > R_T \tag{6}$$

where, R_T is a threshold that decides the falseness of the nearest neighbors.
This criterion works well for clean noise free signals. For instance in the context of thermoacoustic instability, systems involving laminar flames mostly exhibit oscillations that manifest as clean signals which are nearly noiseless. For the analysis of such systems, Kabiraj et al. (32) for the first time used the method of FNN to find the minimum embedding dimension of the acoustic pressure signals obtained at different values of the control parameter. On the other hand, for noisy data, FNN method gives spurious results. For such noisy signals, Abarbanel et al. (24) suggested an another criterion of false neighbors as,

$$\frac{R_n(d+1)}{R_A} \geq 2 \tag{7}$$

where, R_A measures the size of the attractor and is given by $R_A^2 = \frac{1}{N}\sum_{n=1}^{N}\left[x(n) - \bar{x}\right]^2$ and $\bar{x} = \frac{1}{N}\sum_{n=1}^{N}[x(n)$.

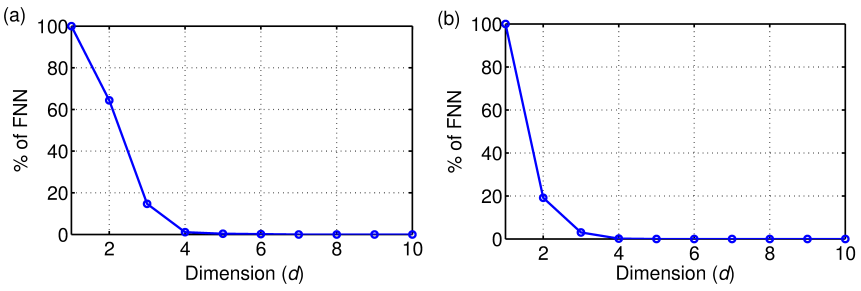

Figure 11

(a), (b) The variation of percentage of false nearest neighbors (FNN) with different values of the embedding dimension (d) for the signals shown in Fig. 2(a) and 2(b) respectively. When the percentage of FNN falls to zero for a particular value of the embedding dimension for the first time, the dimension corresponding to the next higher value is considered as the proper embedding dimension required for the phase space reconstruction. In the case of combustion noise (a) as well as thermoacoustic instability (b), the value of FNN approximately goes to zero at embedding dimension of 4. Thus, the embedding dimension of 5 could be the optimum value of d for the reconstruction of the phase space.

```
function [FNN]=ARFM_fnn(x,Fs,dmax,tao,Rtol,Atol)
% This function calculates the minimum value of embedding dimension required
% for the reconstruction of phases space of a given time sampled data.
% It is based on the concept of false nearest neighbors.
% The main idea is to eliminate the false crossings of the phase space trajectories
    that mainly arise
% due to the projection of higher-dimensional original attractor into a
    lower-dimensional phase space
% The method of FNN measures percentage of closeness, in terms of Euclidian
    distances, of neighboring points
% of the trajectory in a d-dimensional space, and compared it with the d+1
    dimensional space.
```

```
% If the ratio of these distances is greater than a predefined threshold due to change in the dimension,
% the neighbors of the trajectory are considered as false neighbors
% The minimum embedding dimension is chosen as the next dimension to the value where the percentage of
% false nearest neighbors become zero for the first time
%_______________ Inputs _______________
% x:        Time series
% Fs:       Sampling Frequency
% dmax:   Maximum number of dimensions that need to be tested
% tao:      Time-delay required for phase space construction. It is calculated from Average Mutual Information
% Rtol:     Distance threshold to decide false neighbor (Value lies between 10 < Rtol < 50, usually fixed to 10)
% Atol:     Another criterion to remove false neighbors (Fixed value, 2) (Abarbanel et al. 1993)
%_____________________________________
% Require a function "delay_vec.m" to calculate the number of time-delayed vectors using a Takens' embedding theorem
% _____________ Reference ___________
% 1) Kennel, M.B., Brown, R., and Abarbanel, H.D., 1992. Determining embedding dimension for phase-space reconstruction using a geometrical construction. Physical review A, 45(6), p.3403.
% 2) Abarbanel, H.D., Brown, R., Sidorowich, J.J., and Tsimring, L.S., 1993.  The analysis of observed chaotic data in physical systems. Reviews of modern physics, 65(4), p.1331.
% 3) Nayfeh, A.H., and Balachandran, B., 2008. Applied nonlinear dynamics: analytical,  computational and experimental methods. John Wiley & Sons.
%  Example: [FNN]=ARFM_fnn(x,10000,10,3,10,2)
%%_________________________________
% Subtract mean from the signal so that the signal will fluctuate around a zero line
x = x - mean(x);
N=length(x);    % Get length of the signal
% Generate a time vector from the known values of sampling frequency and length of the given signal
delta_t = 1/Fs;
t = 1:N;
t = t'*delta_t;
% The algorithm to detect false nearest neighbors is based on the steps suggested by Kennel et al. (1992)

for k=1:dmax        % Loop to calculate false neighbors for various values of embedding dimension
    N1=N-k*tao;     % Calculating maximum number of delay vectors for given dimension     (d), signal length (N), and time delay (tao); i.e., [N-(d-1)*tao]
    Y=ARFM_delay_vec(x,tao,k,N1);    % Find number of delayed vectors of a signal from the given values of embedding dimension and delay
    FNN(k,1)=0;     % Initializing the value
    for j=1:N1      % Loop to calculate false neighbors of a phase space trajectory at a given embedding dimension

        Y0=ones(N1,1);  % Generate a vector of ones.
        Y1=Y0*Y(j,:);   % Generate the copies of a given vector equal to total number of delayed vectors
        R=sqrt(sum((Y-Y1).^2,2));   % Calculate the distance of a given point of the phase space trajectory with all other points
        [a, b]=sort(R);         % Sort the distances in ascending order, it is necessary to calculate the nearest neighbor
    % Pick second minima of the distances as a first nearest neighbor, as the first minima is always zero - a distance of the point with itself
        NN=b(2);    % Position of the nearest neighbor on the phase space trajectory
        ND=a(2);    % The distance of the nearest neighbor from the given point on the trajectory.......[ Eq. (2) in the paper by Kennel et al. (1992)]
        Rd=(x(j+k*tao)-x(NN+k*tao));    % Calculate the distance of the nearest neighbour with all points in the increase dimensional space
        Rd1=sqrt(Rd.^2+ND.^2);          % The distance of points on the trajectory due to increasing in dimension from d to d+1......[ Eq. (3) in the paper by Kennel et al. (1992)]

    % Condition to check the falseness of nearest neighbors
        if abs(Rd)/ND > Rtol
            FNN(k,1)=FNN(k,1)+1;       % [ Eq. (4) in the paper by Kennel et al. (1992)]
        elseif Rd1/std(x) > Atol
```

```
57                FNN(k,1)=FNN(k,1)+1;        % [ Eq. (5) in the paper by Kennel et al.
    (1992)]
58            end
59        end
60    end
61        % Calculate the percentage of false nearest neighbors for every embedding
    dimension
62    FNN=(FNN./FNN(1,1))*100;
```

Listing 3 Function for calculating embedding dimension (m) through FNN; ARFM_fnn.m

2. *Cao's method:*

As mentioned in the previous description, the method of false nearest neighbor (FNN) is threshold dependent (on the choice of R_T), which means that for different values of R_T, the method will produce different values of the embedding dimensions. To overcome this issue, Cao (33) suggested another method which is an optimization of the previous FNN method.

Consider a time sampled data $x_1, x_2, x_3, \cdots, x_n$. Then, the reconstructed time-delay vectors can be calculated as, $y_i(d) = [x_i, x_{i+\tau}, \cdots, x_{i+(d-1)\tau}]$, where $i = 1, 2, \cdots, N-(d-1)\tau$, and $y_i(d)$ is a i_{th} reconstructed vector in an d-dimensional space.

Now, define a new quantity $a(i,d)$ as follows,

$$a(i,d) = \frac{||y_i(d+1) - y_{n(i,d)}(d+1)||}{||y_i(d) - y_{n(i,d)}(d)||} \tag{8}$$

where, $i = 1, 2, 3, ..., N - d^*\tau$, and $||.||$ is a Euclidian norm of the distance. $y_i(d+1)$ is the i^{th} reconstructed vector in the $d+1$ dimensional space, and $n(i,d)$ is the nearest neighbor of $y_i(d)$ in d-dimensional space. The choice of $n(i,d)$ depends on both i and d. Thus, if d is a proper embedding dimension of the attractor, then any two points in the d -dimensional phase space will always remain close in the $d+1$ dimensional space. Such neighbors are called true neighbors, otherwise, they are termed false neighbors.

The mean of all $a(i,d)$ can be calculated as, $\mathrm{E}(d) = 1/(N-d\tau)\sum_{i=1}^{N-d\tau} a(i,d)$. The quantity $\mathrm{E}(d)$ depends on the dimension d and the time delay τ. The variation of $\mathrm{E}(d)$, as the value of dimension changes from d to $d+1$, is expressed in terms of new quantity $\mathrm{E1}(d)$ as, $\mathrm{E1}(d) = \mathrm{E}(d+1)/\mathrm{E}(d)$.

In this case, $\mathrm{E1}(d)$ will stop changing once all the false neighbors are resolved after a dimension d_0. Then, $d_0 + 1$ can be chosen as the minimum embedding dimension required for the construction of the phase space.

Cao (33) further defined another quantity (called $\mathrm{E2}(d)$) that will help in distinguishing deterministic signal from a stochastic one. If, $\mathrm{E}^*(d) = 1/(N-d\tau)\sum_{i=1}^{N-d\tau} |x_{i+d\tau} - x_{n(i,d)+d\tau}|$, then the value of $\mathrm{E2}(d)$ can be calculated as, $\mathrm{E2}(d) = \mathrm{E}^*(d+1)/\mathrm{E}^*(d)$.

The value of $\mathrm{E2}(d)$ remains always equal to 1 for every value of d, in the case of random signals; whereas, it has values not equal to one for few initial values of d, in the case of deterministic signals.

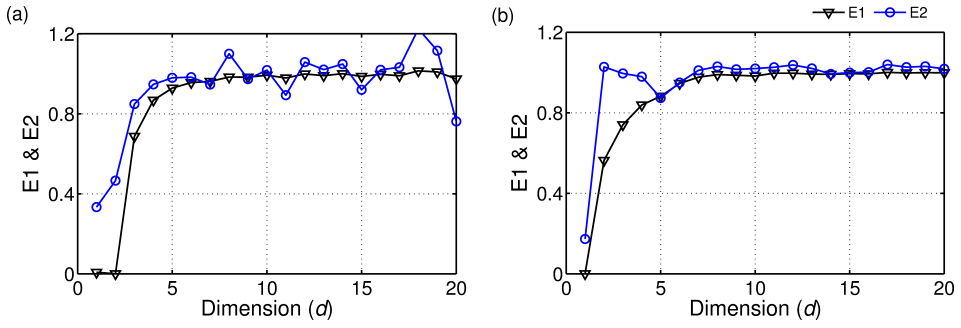

Figure 12

Cao's method used to calculate the minimum embedding dimension required for the reconstruction of the phase space from the acoustic pressure signals obtained during (a) combustion noise and (b) limit cycle state of the oscillations as shown in Fig. 8. During the occurrence of combustion noise state, as the value of $E2$ starts from a value near to 0.4, which implies that the acoustic pressure oscillations during this state may have deterministic behavior. The plots of $E1$ in both (a) and (b) saturate after embedding dimension of 6, thus the dimension of 6 or 7 can be used as a minimum embedding dimension in the construction of phase space.

In the context of thermoacoustic instability, this method of calculating the minimum embedding dimension was for the first time used by Nair et al. (34), for the analysis of the acoustic pressure data obtained from a turbulent combustion system. Furthermore, in order to characterize the deterministic nature of these acoustic pressure signals, especially during a state of combustion noise, various types of other techniques

have been recently used. Nair et al. (34) used the 0-1 test to detect chaos. On the other hand, Tony et al. (35) performed rigorous analysis, by using various measures of detecting chaos such as surrogate analysis, permutation entropy and translation error to confirm the deterministic nature of the data.

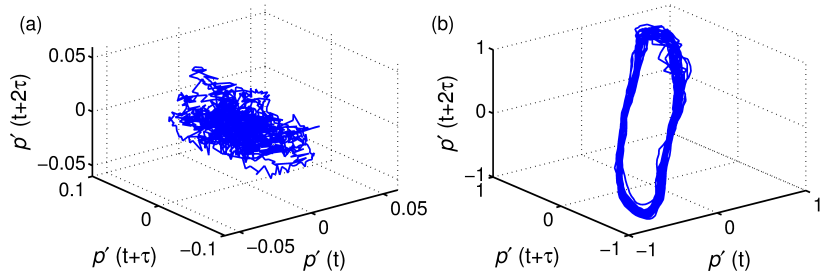

Figure 13

(a), (b) The reconstructed phase plots of acoustic pressure signals obtained during (a) combustion noise and (b) limit cycle states of oscillations. The phase portrait of combustion noise state shows noisy cluttered attractor, whereas, that of the limit cycle state depicts a closed loop structure. The small fluctuations in the closed loop trajectory of the limit cycle attractor could be the result of the presence of noise in the system.

```
function [E1,E2] = ARFM_cao(x,tau,d)
% This code will help in finding minimum embedding dimension using a method proposed by
    Cao (1997)
% _____________ Inputs _____________________
% x:    The time series
% tau:  Optimum time delay obtained from average mutual information (AMI)
% d:    Maximum number of embedding dimensions required for the calculation
% _____________ Reference_________________
% Reference:
% Cao, L., 1997. Practical method for determining the minimum embedding dimension of a
    scalar time series.
% Physica D: Nonlinear Phenomena, 110(1-2), pp.43-50.
% ________________________________________
% We need to get a plot of variation of parameters E1(d) and E2(d) (defined in the
    paper by Cao, 1997) with dimension
% This plot will give the appropriate value of minimum embedding dimension required for
    phase space reconstruction
% In this case, E1(d) will stop changing once all the false neighbors are resolved
    after a dimension d_0.
% Then, d_0+1 can be chosen as the minimum embedding dimension required for the
    construction of the phase space.
%________________________________________
% Initializing the null vectors of E1 and E2 for d+1 dimension
E_m = zeros(d+1,1);
Estar_m = zeros(d+1,1);
% Calculating the values of E1 and E2 for every embedding dimension
for m = 1:d+1
    % Reduce the data vector to the nearest multiple of delay
    n = floor(length(x)/tau)*tau;
    % Store the delayed signals i.e., Y(d)={x(t), x(t+tau)... } in the form of row
    vectors,
    % e.g., for d=2 you have x(t) and x(t+tau), and data length is [n-2*tau]
    Y = zeros(n-m*tau, m);  % Initializing a delay vector, where n-m*tau is the total
    number of delay vectors
    % Loop to calculate delay vectors for a given time series using Takens delay
    embedding theorem (1981)
    for i = 1:m+1
        Y(:,i) = x((i-1)*tau+1:(n-(m+1-i)*tau));
    end

    N = size(Y,1); % Length of the delay vectors
    % Initializing two vectors required to store the values of a(i,d) (Eq. (1) in Cao,
    1997)
    % and distance vector of Eq. (4) in Cao, 1997
    dist = zeros(N,1);
    a2 = zeros(N,1);
    % Loop to calculate the value of a(i, d) for every d as given by equation (1) in
    Cao (1997)
    for i = 1:N

        temp = Y(i,1:end);  % Swapping of Y vector after each iteration
```

```
        Y(i,1:end) = Y(1,1:end);
        Y(1,1:end) = temp;
        % Obtain a distance matrix of points on the phase space trajectory
        % i.e., calculating the distance of every initial reference point on the phase space trajectory
        % with respect to other points of trajectory and repeating the same process for N-1 points
        Rd = transpose(sqrt(sum((repmat(Y(1,1:end-1),size(Y,1)-1,1)...
            - Y(2:end,1:end-1)).^2,2)));
        % Finding the nearest neighbor to a given reference point of the phase space trajectory i.e., finding the
        % value of minimum distance from the distance matrix and the position of this neighbor from the reference point
        [val_mRd, pos_mRd] = min(Rd);
        % Obtain the distance nearest neighbor with other points on the trajectory due to increasing the dimension by 1.
        % This is needed for the calculation of E2
        dist(i) = abs(Y(1,end) - Y(1+pos_mRd,end));
        % Avoid division by 0
        val_mRd = val_mRd + eps;
        % Obtain the total distance on increasing the dimension by 1
        minRdplus1 = sqrt(val_mRd*val_mRd + dist(i)*dist(i)) + eps;
        % Take the ratio of distances. This is needed for E1
        a2(i) = minRdplus1/val_mRd;     % Equation (1) in the paper by Cao (1997)
    end
       E_m(m) = mean(a2);          % Equation (2) in the paper
       Estar_m(m) = mean(dist);    % Equation (4) in the paper by Cao (1997)
end
      % Finding E1(d) and E2(d) for every dimension
      E1 = E_m(2:end)./E_m(1:end-1);              % Equation (3) in the paper by Cao (1997)
      E2 = Estar_m(2:end)./Estar_m(1:end-1);      % Equation (5) in the paper by Cao (1997)
end
```

Listing 4 Function for calculating embedding dimension (m) through Cao's method; `ARFM_cao.m`

4. Poincaré Section (First Return Map):

A Poincaré section or map explores the properties of a continuous time system by lowering its phase space dimension. This is achieved by transversely cutting an m dimensional attractor of the system through an $m-1$ dimensional plane. This technique was first discovered by Henri Poincaré (36). It reveals the various types of motion such as periodic, quasiperiodic and chaotic motions exhibited by the given dynamical system (8). The use of this technique requires a careful selection of the plane of intersection. This ensures the proper detection of the existing dynamics in the system, and reduces the possibility of having ambiguity in the analysis.

For systems whose asymptotic behavior is limited to n-dimensional space, where $n \leq 3$, a first Poincaré map is sufficient to ascertain the dynamics of this system. On the other hand, if the system is higher dimensional ($n > 3$), we require a 2 or 3-dimensional projection of the Poincaré section, which is mainly required for systems involving higher-dimensional quasiperiodic or chaotic motions (8). The presence of crossing (or intersection) of the phase space trajectory in the Poincaré section (say, in 2-D) concludes that the projection should be of at least one-dimensional larger (3-D or higher) to resolve the dynamics of that particular system.

In the case of periodic systems of known period T, the first Poincaré map (also known as the first return map) would produce N discrete points in the two-dimensional phase space. Here, N could be an integer with a value equal to 1 for periodic oscillations, 2 for period-2 oscillations, and so on. If N is a non-integer but rational number, the motion is called mode-locked, whereas, if the N is irrational, then the motion is called quasiperiodic. For the mode-locked (phase-locked) state, the dynamics of the system recurs after a time period decided by the combination of two rational integer numbers. The first Poincaré section for such systems will show a set of discrete points in the intersection plane. In the case of quasiperiodic motion, the system dynamics is aperiodic, which is a result of two or more incommensurate frequencies (i.e., the frequency ratio is an irrational number and hence the motion of the system does not repeat ever), the first Poincaré section shows a closed dense circle in the two-dimensional subspace. The Poincaré sections of chaotic or higher-dimensional quasiperiodic systems do not exhibit a simple geometric pattern, as observed for the periodic or 2-period quasiperiodic motions (8). Therefore, the Poincaré section is not a good method to characterize such high dimensional systems. If the Poincaré section does not show any fixed point or a closed loop, the underlying dynamics could be considered as chaotic.

```
function [pks]= ARFM_Return_Map(x,Fs)
```

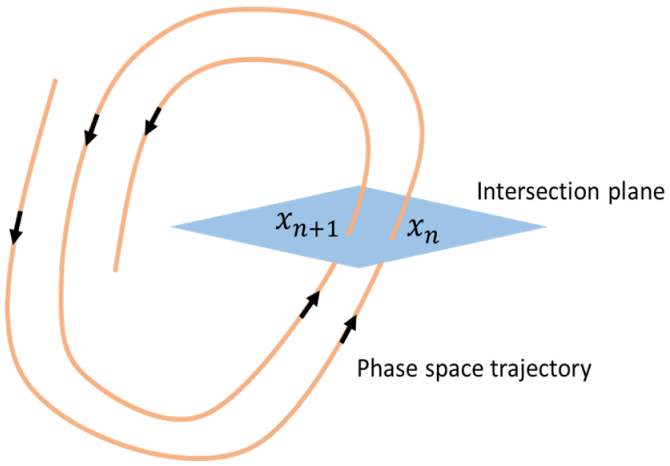

Figure 14

Schematic of the construction of a Poincaré map by transversely intersecting the attractor of the system dynamics in a given direction so that it reveals the original dynamics of the system. In the figure, x_n is the first intersection and x_{n+1} is the second intersection of the phase space trajectory with the Poincaré plane. The plot of x_{n+1} versus x_n is referred to as the first return map of the system dynamics.

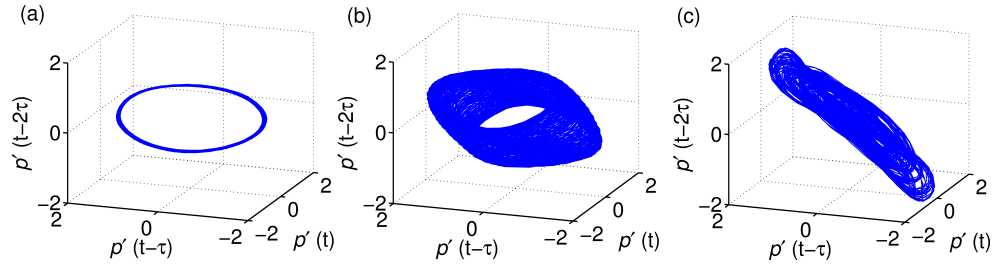

Figure 15

(a) - (c) The time evolution of the phase space trajectory of an acoustic pressure signal, in the 3-D reconstructed phase space, obtained during period-1 (limit cycle), quasiperiodic, and chaotic states of oscillations respectively from a laminar combustor. These three states of system dynamics correspond to three different locations of the flame inside the combustor, wherein the movement of the flame location along the duct length was the control parameter. For the period-1 state, the phase space shows a closed ring type structure, whereas, for the quasiperiodic state, the phase space depicts a toroidal structure. In the case of the chaotic state, the phase space shows a strange attractor, wherein the evolution of the trajectories is apparently random. The parameters chosen for plotting the phase space are a time delay (τ) = 8, and an embedding dimension (d) = 6 (The data is published in ref. (19)).

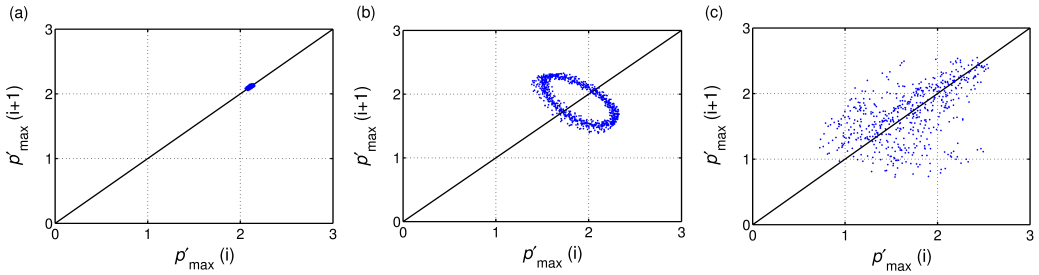

Figure 16

(a) - (c) The first return plots of the system dynamics corresponding to all the states of acoustic pressure oscillations shown in Fig. 2 respectively. The first return map is a plot between the first local maxima of the signal with respect to its next local maxima. For the period-1 (limit cycle) state (a), the first return map shows a single point, whereas, for the period-2 quasiperiodic oscillations (b), it shows a dense closed loop structure. In the case of chaotic oscillations (c), the first return map demonstrates the apparently random distribution of points without any regular pattern (The data is published in ref. (19)).

```
% This code will help in exploring the properties of the higher dimensional system into
    a lower dimensional subspace.
% This method is also known as Poincar  section
% It helps in revealing the various types of motion such as periodic, quasiperiodic and
    chaotic exhibited by the given dynamical system
% In the case of periodic systems of known period T, the first return map would produce
    N discrete points
% in the two-dimensional phase space. Here, N could be an integer with a value equal to
```

```
    1 for periodic
% oscillations, 2 for period-2 oscillations, and so on.
% If N is a non-integer but rational number, the motion is called mode-locked, whereas, if the N is irrational,
% then the motion is called quasiperiodic.
% For the mode-locked state, the first return map will show a set of discrete points in the intersection plane.
% For quasiperiodic motion, the first return map shows a closed dense circle in the two-dimensional subspace.
% The Poincar  sections of chaotic or higher-dimensional quasiperiodic systems do not exhibit a simple
% geometric pattern, as observed for the periodic or 2-period quasiperiodic motions
% In order to create first return map, (1) the extreme events (such as local maxima or minima or zero crossing)
% of every cycle of the signal is calculated and (2) then the first event is plotted against the next one.
% ______________ Input _____________________
% x:    Time series
% Fs:   Sampling frequency
% ____________Reference _____________________
% Nayfeh, A.H., and Balachandran, B., 2008. Applied nonlinear dynamics: analytical,
%       computational and experimental methods. John Wiley & Sons.

% E.g.: [pks]= ARFM_Return_Map(x,10000)
%% Get first return map

% Calculate all positive maxima's of every cycle of the given time series
% Here, pks - values of local maxima's and locs - time indices corresponding to these maximum values
[pks,locs]=findpeaks(x,'MinPeakDistance',10,'Minpeakheight',0);

figure(305);
set(gcf,'Position',[550,450,800,500])
% set(gcf,'Position',[300,300,1400,500]);
% subplot (121)

% Create a reference diagonal line in the plot with size sufficiently higher than maximum value of the signal
M=max(pks)+2*max(pks);
plot(0:1/Fs:M,0:1/Fs:M,'linewidth',1.5,'Color', [0 .5 0]);  grid on;  hold on;

% Plot a first return map wherein the first maxima of the signal is plotted with the next maxima
plot(pks(1:end-1),pks(2:end),'.k','Color', [.8 .1 0],'MarkerSize',14);
    set(gca,'fontsize',18)

axis tight
title('First return map','fontsize',22,'Color', 'b','Fontweight','bold')
xlabel('\itx\rm_{max} (i)','fontsize',24)
ylabel('\itx\rm_{max} (i+1)','fontsize',24)
xlim([0,M]);
ylim([0,M]);
set(gca,'linewidth',1.5)
```

Listing 5 Function for plotting first return map; `ARFM_Return_Map.m`

5. Recurrence Plot Analysis:

The recurrence phenomenon is ubiquitous in nature (37). It is a property of deterministic dynamical systems (38)). Recurrence refers to the phenomenon of the phase space trajectory revisiting the neighborhood of its previous location, after a time of divergence in the phase space. With the use of this property, the time evolution of the system dynamics can be qualitatively captured by plotting the recurrence plot (RP). Recurrence plots help in depicting certain aspects of a higher-dimensional attractor into a two-dimensional subspace. The primary use of recurrence plots is not only to visualize the time series but also to detect hidden complex patterns present in the data (38). The advantage of RP is that it requires short time series data to characterize the dynamical behavior of the signal, and can be successfully used for nonstationary signals (39).

The first step in constructing a recurrence plot is to embed the time series of a given dynamical system into an appropriate low-dimensional phase space using the Takens delay embedding theorem (26). The parameters required to reconstruct the phase space are the optimum time delay (obtained from the method of average mutual information (31)) and the minimum value of embedding dimension (obtained from the false nearest

neighbor method (25) or Cao's method (33)). In addition to these, another important parameter required for the construction of recurrence plot is the cut-off threshold (40). The size of the threshold depends on the applicability of RP to a particular type of problem (41). It can be chosen as the fraction of the size of the attractor or can be considered as a fixed number of nearest neighbors. The increase in the size of the threshold correspondingly increases the number of recurrences in the RP. The equation used in the construction of the recurrence plot is given by,

$$R_{i,j} = \Theta(\epsilon - ||\vec{x}_i - \vec{x}_j||); \qquad i, j = 1, 2, \cdots, N_1 \tag{9}$$

where, $\vec{x}_i$ and $\vec{x}_j$ are the state space vectors, Θ is the Heaviside step function (i.e. $\Theta(X) = 0$ if $X < 0$, and $\Theta(X) = 1$ if $X \geq 0$), $N_1 = N - (d-1)\tau$ is the total number of points in the delayed vector, ϵ is a predefined threshold, and $||\cdot||$ is the Euclidian norm. Whenever the phase space trajectory falls within the threshold, we mark it as 1 in the recurrence matrix; otherwise, we mark it as 0. Thus, the recurrence plot is a graphical representation of black and white points, where a black point corresponds to $R_{i,j} = 0$ and a white point corresponds to $R_{i,j} = 1$.

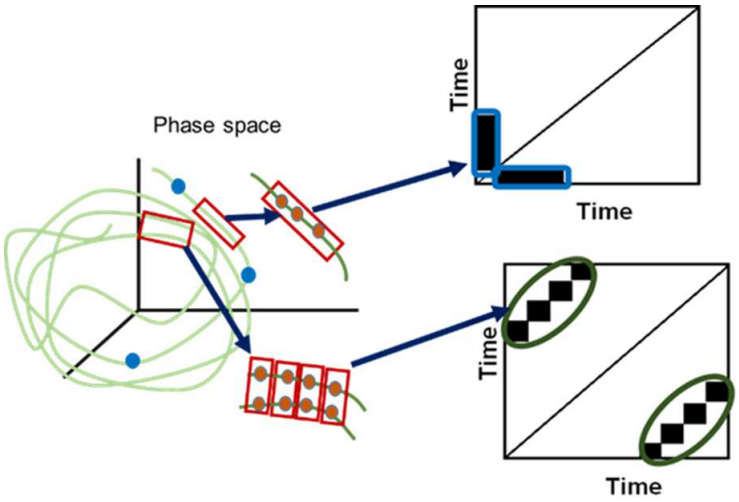

Figure 17

Schematic of the construction of recurrence plot (RP) based on the recurrence behavior of the phase space trajectory. The diagonal lines in RP depict that trajectories are running parallel to each other, whereas, the vertical lines in RP demonstrate the recurrence (or trapping) of more than one point of the same trajectory inside the predefined threshold.

As mentioned earlier, the structural patterns present in the recurrence plots characterize the dynamical behavior of the phase space trajectory of a given system. The long continuous lines parallel to the main diagonal line illustrate the presence of periodic process, in which, the separation distance between the consecutive lines is equal to the period of the signal (38). Homogeneously distributed black points in the RP indicate a white noise (random) process (39). For such signals, the isolated black points indicate that the evolution of the phase space trajectory is unpredictable at the next instance of time. On the other hand, in the case of a chaotic process, the phase space trajectory diverges after a few time steps of closeness, the RP of such signals show apparently random distribution of short diagonal lines with a few single points in the RP (39). When the system dynamics consists of both fast and slow varying scales, the dynamical evolution of such signals in the RP results in the appearance of large patches of black and white points. In this, the black patches consist of vertical lines representing the trapping of slow varying oscillations in between the fast evolving dynamics. In the dynamical systems literature, such slow varying states are called laminar states (or intermittency) (42). The parallel lines orthogonal to the main diagonal line indicate the insufficient embedding of attractor in the phase space (39).

Kabiraj and Sujith (32), for the first time, applied RP technique to thermoacoustic systems. Using this method, they distinguished various kinds of system dynamics such as periodic, quasiperiodic, chaotic, and intermittent oscillations observed in the acoustic pressure signals of a laminar combustor for particular values of the control parameter. They further used recurrence plots to characterize the type of intermittency witnessed in their system prior to a blowout of the flame. Likewise, Nair et al. (43) used RP to differentiate different types of dynamics observed during intermittency route to thermoacoustic instability in a turbulent combustor. They noticed that RP during combustion noise shows grainy structures of black and white points, whereas, that during thermoacoustic instability displays parallel diagonal lines. During intermittency, the RP plots show large size black patches in the background of white regions. Thereafter, numerous researchers have successfully used the approach of RP to understand different aspects of the system dynamics of the thermoacoustic systems (44; 45; 46; 47; 48; 49; 50).

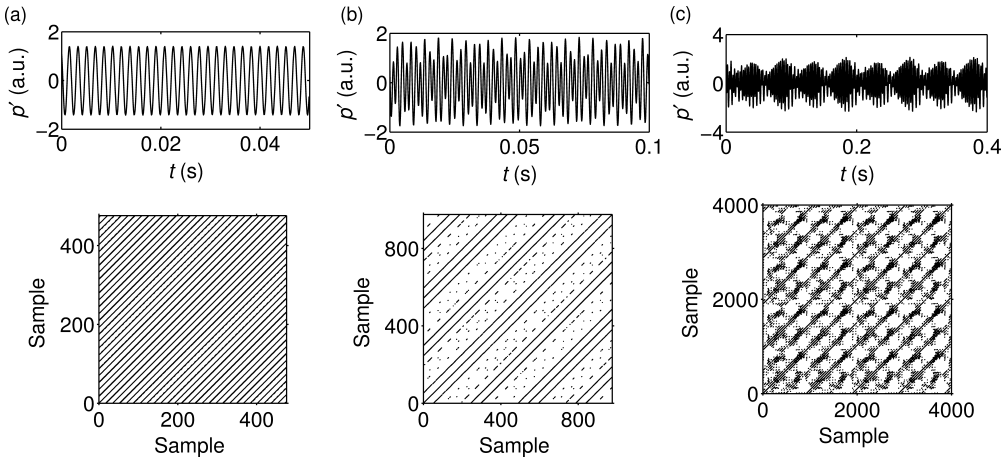

Figure 18

(a) - (c) The time series and corresponding recurrence plots of the acoustic pressure signals obtained during the limit cycle, quasiperiodicity and chaotic state of oscillations respectively, from a laminar premixed flame combustor at different values of the control parameters. During the limit cycle state, the RP shows diagonal lines, where the separation distance between the consecutive diagonal lines is the time period of the oscillations. For quasiperiodic oscillations, RP shows diagonal lines with irregular spacing, manifesting the irrational relation of the instability frequencies. In the case of chaotic oscillations, RP depicts the broken short diagonal lines representing the characteristics of the chaotic oscillations as the neighboring trajectories diverge after a short time of closeness in the phase space (The data is published in ref. (19)).

```
function [S]=ARFM_Recurrence_plot(x,Fs,d,tau,threshold)
% This code will display recurrence plot of a given time series
% In order to get recurrence plot, we first need to reconstruct a phase space using appropriate
% values of time delay and embedding dimension. After the reconstruction of phase space, the distance
% every point of the trajectory is calculated with respect to other points to create a distance matrix.
% Using an appropriate value of threshold (percentage of the maximum size of the attractor), a recurrence
% matrix is created. A two-dimensional plot of the recurrence matrix is called recurrence plot.
% It is better to use less data point in order to reduce computation time (e.g. N=1000 points)
% _________________ Inputs ____________________
% x:    Time series
% Fs:   Sampling frequency
% d:    Minimum embedding dimension
% tau:  Optimum time delay (in samples)
% threshold:    Threshold to detect recurrences (e.g. 10%, 20%, etc.)
% ________________ Output _________________
% S:    Distance matrix
% e.g. [S]=Recurrence_plot(x,10000,6,20,20);
% ________________ Reference _____________
% N. Marwan, M. Carmenromano, M. Thiel, and J. Kurths,
%       "Recurrence plots for the analysis of complex systems", Phys. Rep. 438, 237 (2007).

%% Plot the signal
N=length(x);            % Calculate length of the signal

if N > 1000
    disp(1,'Please reduce the number of data point to avoid computational delay');
        return;
else
    disp('Number of data points are sufficient to run the code smoothly');
end

x = x - mean(x);        % Calculate the mean subtracted signal

% Generating a time vector of the sampled data using a given value of sampling frequency (Fs)
delta_t = 1/Fs;   t = 1:N;  t = t*delta_t;

```

```
% Plot the signal
figure(306);
set(gcf,'Position',[200,300,1600,700]);

subplot(1,2,1);
plot(t,x,'k','linewidth',1.5);
title('The signal','Color', 'b','fontsize',20,'Fontweight','bold');
xlabel('Time (s)','fontsize',20)
ylabel('Amplitude','fontsize',20)
set(gca,'linewidth',1.5); set(gca,'fontsize',18)
set(gca,'Position',[0.1 .25 0.4 0.6])
axis tight
%% Construction of delay vectors using Taken's embedding theorem

% Find length of the delay vectors (M) using known values of number of data points in the signal (N),
% minimum embedding dimension (d), and optimum time delay (tau) as: M=N-(d-1)*tau

M=N-(d-1)*tau;

% Initializing a matrix of delay vectors
Y = zeros(M,d);

% Loop to get time delayed vectors using a time delay embedding theorem of Taken's
for i=1:d
    Y(:,d-i+1) = x(1+(i-1)*tau:M+(i-1)*tau);
end

%% Get distance matrix

L1 = length(Y(:,1));   % Length of the column delay vecor

% Creating a distance matrix by calculating the distance of every point of phase space trajectory
% with other points, and repeating the same process for all other points

x1 = repmat(Y, L1, 1); % Replicating the M1 matrix L1 times

x2 = reshape(repmat(Y(:), 1, L1)', L1 * L1, d);  % Arranging delay vector matrix in such a way that every row of the matrix is repeated for L1 times


S = sqrt(sum( (x1 - x2) .^ 2, 2 ));         % Subtraction of x1 and x2 will create a distance vector
S = reshape(S, L1, L1);                     % Reshaping the distance vector into a distance matrix form

%% Get recurrence plot

% Creating the time vector of length L1 (i.e., number of points in the column of a distance matrix)
t1 = t(1:L1);
t2 = t1;

% Recurrence threshold can be given in terms of percentage of maximum size of the attractor (i.e., maximum value of distance from the distance matrix)
% We can use another threshold such as the percentage of Euclidian distance or fixed number of nearest neighbors etc. [Ref: Marwan (2007)]
eps=threshold/100;   % Set given recurrence threshold in percentage

subplot(1,2,2);
imagesc(t1,t2, S < eps*max(S(:)) )  % Creating a Recurrence matrix of 0's and 1's from the distance matrix by using a proper value of recurrence threshold
axis square
set(gca,'YDir','normal')

colormap([1 1 1; 0 0 0]) % Converting color plot into a plot of black and white points

title('Recurrence Plot', 'Color', 'b','fontsize',20,'Fontweight','bold');
xlabel('Time (s)','fontsize',20)
ylabel('Time (s)','fontsize',20)
set(gca,'linewidth',1.5); set(gca,'fontsize',18)
set(gca,'Position',[0.6 .25 0.3 0.6])
```

Listing 6 Function for producing recurrence plot for a given time series; ARFM_Recurrence_plot.m

6. Complex Networks (Visibility Algorithm):

One of the recent advancements in nonlinear dynamical systems theory is the construction of complex network from a given time series of the sampled data (51). Various methods have been devised over the years to achieve this, some of them are transition networks, cycle networks, correlation networks, visibility graphs, k-nearest neighbor networks, or recurrence networks (52). Among these techniques, we use a method based on the visibility algorithm suggested by Lacasa et al. (53). In this method, every point of the time series is considered as a node. The two nodes of the signal are connected only if there is a possibility of drawing a straight line connecting these nodes which is not intersected by the other nodes present in between the two nodes. Such connecting line is also known as visibility line and the condition of joining two nodes of the signal is called visibility condition.

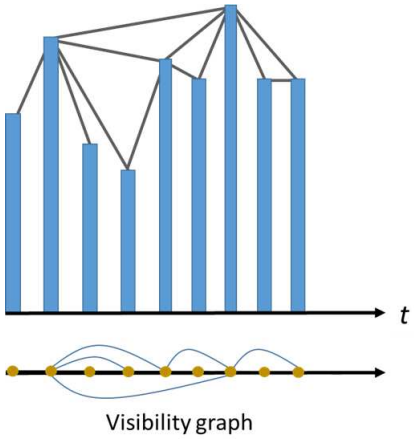

Figure 19

Schematic of the visibility graph algorithm in which each vertical bar represents the magnitude of every point (node) of the sampled signal. The connection between the neighboring nodes (links) is based on the satisfaction of the visibility condition of the node. The schematic of the connectivity between the nodes is shown in the bottom figure.

Let us consider two nodes, say i and j, of an evenly sampled signal x separated by another node k such that $t_i < t_k < t_j$, the connectedness of these nodes is decided by the following criterion of visibility (53):

$$x_k < x_j + (x_i - x_j)\frac{t_j - t_k}{t_j - t_i} \tag{10}$$

Since the condition of visibility is symmetric for both nodes i and j, the networks constructed from this technique are called undirected networks (54). It means that the information about directionality (in or out) of the connections (also known as links) between the nodes is not provided while constructing the network. The nodes are connected, which means they have visibility for their neighbors (left or right) (53).

Once the condition of visibility is satisfied, an adjacency matrix (A_{ij}) is computed from the information of connectedness of the nodes. The adjacency matrix (for undirected networks) is a symmetric matrix in which the non-diagonal elements of A_{ij} represent the connection between nodes i and j. If the two nodes of the signal are connected, their connection is represented by 1 in A_{ij}, otherwise, it is marked as 0 in A_{ij}. Self-connections of the nodes, i.e., points along the diagonal line of A_{ii} are marked as 0. Thus, A_{ij} provides the total information required for the construction of complex network from a given time series. The connections between the nodes are quantitatively computed from the measure known as the degree of the node (53). The degree gives the information about how many nodes of the signal are connected to a given node and, is calculated as $k_i = \sum_{j=1,i\neq j}^{N} A_{ij}$. The percentage of the nodes with degree k are quantified in terms of their degree distribution $P(k)$ and, is given by $P(k) = n_k/n$. Here, n is the total number of nodes in the network, and n_k is the number of nodes with degree k. The degree distribution is one of the important properties that characterizes the structure of the network (55). It gives an idea about the spread of the number of links that every node of the network possesses. In the network terminology, the node with the maximum degree is called the hub of the network.

Various types of structural patterns arise due to the complex interaction between the constituents (nodes) of the system. According to the distribution of links (connectedness) between the nodes, complex networks are mainly classified into four types as, i) regular networks, ii) random networks, iii) small-world networks, and iv) scale-free networks.

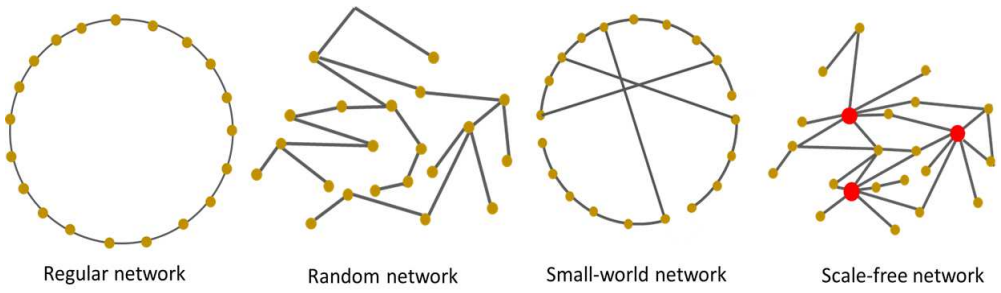

Figure 20

Different types of complex networks. The classification of the types of networks depends on the arrangement of neighboring nodes and the arrangement of connections between them.

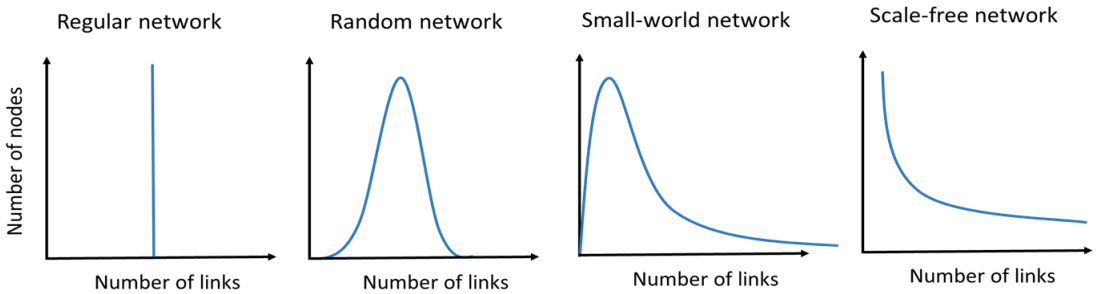

Figure 21

Schematics of degree distributions for different types of networks. For the regular network, it is a Dirac delta function; for the random network, it is a bell-shaped type curve; for small-world networks, the distribution is like Poisson behavior and; for the scale-free network, it is power law behavior.

- **Regular networks:** In these networks, each node of the network is equally connected to a few of its nearest neighbors. Every node of these networks will have the same degree. The degree distribution of regular network is a constant vertical line showing features of the Dirac-delta distribution. As a consequence, regular networks have the highest clustering coefficient as well as the highest characteristic path lengths (56). The clustering coefficient quantifies the probability with which neighbors of a particular node are connected to each other (57). The higher the clustering coefficient, the more robust the network is. On the other hand, the characteristic path length quantifies the average shortest path between any two nodes of the network.

- **Random networks:** Random networks have the opposite network properties of those of regular networks (56). Such networks are constructed by randomly connecting disconnected pairs of nodes of the network through a link selected from the uniform probability distribution. The degree distribution of such a network looks like a bell-shaped curve, in which the shape of the network falls exponentially from both sides of its average value. Random networks have very low values of average shortest path length and the value of the clustering coefficient is close to zero. Such networks are also known as Erdös-Rényi random network (58).

- **Scale-free networks:** These are the networks whose degree distribution retains a power law behavior in the log-log plot of $P(k)$ versus k (59). For scale-free networks, $P(k) = k^{-\gamma}$, where γ is a slope of the asymptotic part of the degree distribution, whose values typically range from 2 to 3. The power law behavior indicates that any part of the network has apparently similar structure to that of the entire network. In scale-free networks, many nodes have only a few connections and a few nodes have many connections. Such behavior of the network results in a long tail degree distribution. Hence, there is no specific scale associated with scale-free networks, and the observation of such behavior holds for both local as well as global scales of the network. Some popular examples of scale-free networks are the world wide web, power grids, and airline networks.(60).

- **Small-world networks:** Most real world networks do not possess the properties of extreme networks such as a regular network or a random network. There is another type of network that falls in between the categories of regular networks and random networks, called small-world networks. These networks were first proposed by Watts and Strogatz (57). Regular networks have the property of large clustering

coefficients, whereas random networks have the property of very low values of average shortest path length. Small-world networks have both properties of large clustering coefficients (as for a regular network) and small average path lengths (as for random networks). Such small-world networks inherit the properties of popular phenomenon of six degrees of separation (61). Some examples of small world networks are the world wide web, power grids, and cellular phone networks(57).

The procedure devised by Watts and Strogatz (57) to construct scale-free networks is as follows. As an example, we start with a regular network on a ring-shaped lattice, wherein each node of the network is connected to its first and second nearest neighbor. Now, first choose any one of the nodes of this ring lattice, then disconnect one of its links and reconnect it back to any randomly chosen node with a probability p. Such random rewiring of the links imposes randomness in the original regular network. The rewiring of links of a randomly chosen node, with a probability $0 < p < 1$, changes the average path length of the network (here $p = 0$ is a regular network and $p = 1$ is a random network). As the value of p increase from 0, the average path length of the network decreases rapidly. The degree distribution of a small-world network shows a peak around the average degree and displays a rapid fall on both sides of the peak.

Murugesan and Sujith (62) for the first time used the complex network approach (visibility algorithm) to study the transition of a turbulent thermoacoustic system from a stable operation to an unstable operation. They discovered that combustion noise (low amplitude aperiodic oscillations) have scale-free behavior that disappears during the onset of thermoacoustic instability (large amplitude periodic oscillations). Furthermore, Okuno et al. (63) used other approaches of complex network analysis such as cyclic networks and phase space networks to characterize thermoacoustic instability. With this, they showed that thermoacoustic instability possesses the property of high-dimensionality with small-world like and power-law type behavior.

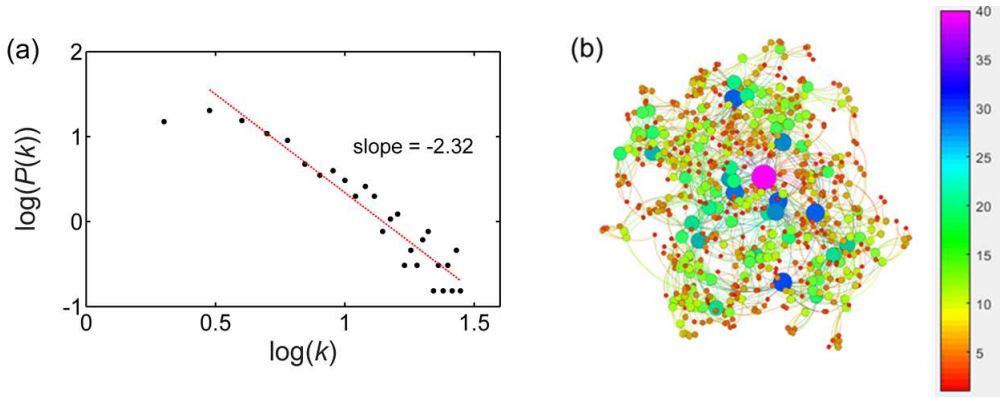

Figure 22

(a) The degree distribution plot of the acoustic pressure signal obtained during a combustion noise state from a bluff-body turbulent combustor. The power law behavior of the tail portion of the degree distribution suggests that the signal is scale-free. (b) The complex network generated using Gephi software (https://gephi.org/) representing the distribution of various nodes of the acoustic pressure signal. In the network diagram, each circle corresponds to a node of the signal (local positive maxima) and connections between nodes show their links. The size of the node represents the corresponding degree of that node.

```
function [deg]=ARFM_Visibility_Algorithm(x,Fs)
% This code extracts the complex network for any given signal using a methode of
    "Visibility Graph Algorithm" (Lacasa et al. 2008)
% This code will help in plotting the degree distribution and getting an adjacency
    matrix required for
% generating a complex network (using Gephi software) of any given time series
% _____________ Inputs _____________________
% x:    Time series
% Fs:   Sampling frequency
% e.g. [deg]=ARFM_Visibility_Algorithm(x,Fs);
% ___________  Reference ________________
% Lacasa, L., et al. "From time series to complex networks: The visibility graph."
%          Proceedings of the National Academy of Sciences 105.13 (2008): 4972-4975.
%%
N=length(x);            % Calculate length of the signal
x = x - mean(x);        % Calculate the mean subtracted signal
% Generating a time vector of the sampled data using a given sampling frequency (Fs)
delta_t = 1/Fs;   t = 1:N;  t = t*delta_t;
figure(506);
set(gcf,'Position',[550,450,700,500])
```

```
% figure('Position',[150,150,1650,700]);%
% subplot(2,2,1);     % Plot the signal
% plot(t,x,'k','linewidth',1.5,'color',[1 .5 0]);  set(gca,'fontsize',18)
% title('The signal','Color', 'b','fontsize',20,'Fontweight','bold');
% xlabel('Time (s)','fontsize',20)
% ylabel('Amplitude','fontsize',20)
% set(gca,'linewidth',1.5)
% set(gca,'Position',[0.1 .25 0.4 0.6])
% Obtain vector of local maxima of a given signal. Each maximum is considered as a 'node' in the complex
% network and the line connecting each node is called 'connection'
% Find magnitude and location of all the local peaks of a given signal
[pks,locs] = findpeaks(x,'MinPeakHeight',0,'MinPeakDistance',20);
peak=pks';          % Obtain a vector of nodes from the signal
% Obtain adjacency matrix. It is required to store the information about the connectivity between the nodes (Donner et al. 2010)
% If two nodes are connected then it is represented by '1' in the adjacency matrix and otherwise, it is referred as '0'
% Two peaks are connected if they are visible    ### For 'visibility condition' read: L. Lacasa et al. 2008,
% The two nodes are connected only if "there exists a straight line
% that connects these nodes and this line is not intersected by any of the intermediate points"

L1=length(peak);    % Find total number of nodes in the signal

A=zeros(L1);        % Defining a adjacency matrix

for i=1:L1-1
    for j=i+1:L1
        %         if
        %
    peak(i+1:j-1)+0.24*meanpeak<(peak(j)+(peak(i)-peak(j))*(j-(i+1:j-1))/(j-i))

        if peak(i+1:j-1)<(peak(j)+(peak(i)-peak(j))*(j-(i+1:j-1))/(j-i))    % Condition of visibility..... [Eq. (1) in Lacasa et al (2008)]

            A(i,j)=1;   % If the visibility condition holds, represent this connection by 1; otherwise, it will be automatically taken as 0
                        % To avoid self-connection A(i,i) is always considered as zero
        end
    end
    A(i,i+1)=1;         % Condition representing the connection between neighboring nodes
end

% The final adjacency matrix will be an upper triangular matrix

A=A+A';     % Generate a symmetric matrix of A

%%  Analyzing complex networks using Gephi software

% In order to analyze signal using Gephi software, first we need to create
% a matrix whose first row and the first column will be numbered from
% 1,2,3,..... to the length of the adjacency matrix.
%This can be achieved by shifting the adjacency matrix in both row as well as column by one

B(2:length(A)+1,2:length(A)+1)=A;         % Shift the Adjacency matrix by one row and one column
B(1,2:length(A)+1)=1:length(A);           % Number the first column, other than first element, as 1 to length of the Adjacency matrix
B(2:length(A)+1,1)=1:length(A);           % Number the first row, other than first element, as 1 to length of the Adjacency matrix
csvwrite('D.csv',B);                      % This creates an excel file containing information of connection of all the nodes of a signal.
                                          % Use this file directly in the Gephi software to create a complex network

%% Find the degree distribution of a complex network mapped from the time series

% Degree of a node: sum of all the nodes that are connected to a given node
deg=sum(A);   % Sum all the numbers of adjacency matrix in column wise i.e., the sum all the number of connections to a particular node
```

```
77
78 table=tabulate(deg(2:L1-1)); % creates a frequency table of the data present in 'deg'
       vector
79 % This command arranges the data in a table as:
80 % 1st column     a vector containing unique values (no repetition) of a 'deg' vector
       arranged in ascending order
81 % 2nd column     The number of instances values in 1st column are repeated in a 'deg'
       vector
82 % 3rd column     The percentage of each value in the total set of degree of a node
83
84 % assigning each column of a table by different variable names
85 value=table(1:end-1,1);
86 count=table(:,2);
87 Percent=table(1:end-1,3);
88
89 x=find(Percent);            % Find the indices of a percentage vector having nonzero
       elements
90 value=value(x);             % Get elements from a 'value' vector having non-zero
       percentage
91 Percent=Percent(x);         % Get elements from a 'percentage' vector having non-zero
       percentage
92
93 % Plot degree distribution of complex network
94 % This is log-log (to the base 10) plot between the percentage of nodes with k number of
95 % connections and the degree of the node
96
97 logvalue = log10(value);
98 logPercent = log10(Percent);
99
100 % subplot(2,2,2);
101 plot(logvalue,logPercent,'ok','MarkerSize',8)
102
103 LV=logvalue(2:end); LP=logPercent(2:end);
104 C=polyfit(LV,LP,1);  % Linear fit to the degree distribution plot
105 Lin_fit=C(1)*LV+C(2);
106 hold on;
107 % C(1)
108 plot(LV,Lin_fit,'r-.','MarkerSize',10,'linewidth',1.5)
109 set(gca,'fontsize',18,'linewidth',1.5)
110 title('Degree distribution', 'Color', 'b','fontsize',20,'Fontweight','bold');
111 xlabel('Log(k)','FontSize',20)
112 ylabel('Log(\itP\rm(k))','FontSize',20)
113 text(1.2, 1.0,sprintf('Slope = %0.2f',C(1)), 'horizontalAlignment',
       'center','FontSize',18)
114 % set(gca,'Position',[0.6 .25 0.3 0.6])
```

Listing 7 Function for constructing a complex network for any given signal using a method of"Visibility Graph Algorithm"; ARFM_Visibility_Algorithm.m

7. Multifractal Detrended Fluctuation Analysis (MFDFA):

A method of multifractal detrended fluctuation analysis has been successfully used over the years in detecting the scaling properties of fractal signals as well as in revealing their long range correlations (64). The process of long-range correlation refers to the slow decay of statistical dependence between the points of a signal with an increase in the time interval between them. The decay of such correlations is much slower than exponential decay. This slow decay of correlation always obeys a power law behavior. The statistical scaling exponent that quantifies this power law is referred to as the Hurst exponent (65). In the case of a monofractal signal, the scaling values always remain same at different levels of resolution of the signal, i.e. monofractal signals always have a single value of the Hurst exponent. However, in other cases, where the signals are more complicated involving different scales at different time intervals, known as multifractal signals, the analysis of such signals requires the calculation of different scaling exponents (66). Hence, the full scaling behavior of such signals cannot be represented by just a single value of the exponent but requires a spectrum representation of the exponents.

The multifractality in a time series may arise i) due to broad probability density function (fat-tail distribution) for values of the time series, or ii) due to the long range correlation of both short as well as large-scale fluctuations, or iii) due to the combination of both (66).

One of the ways to analyze multifractal signals is to use an algorithm of Multifractal Detrended Fluctuation Analysis (MFDFA). The important steps involved in finding the multifractal spectrum of such signals using MFDFA are given below (66; 67).

- **Step - 1:** Consider a time sampled data x_k of length N, where $k = 1, 2, \cdots, N$, whose mean value is $< x >= \frac{1}{N}\sum_{k=1}^{N} x_k$. Then the cumulative sum (profile) of the time series is calculated as $Y(i) = \sum_{k=1}^{N}[x_k - < x >]$

- **Step - 2:** Then divide the newly generated signal, $Y(i)$ into N_s equal number of non-overlapping segments of length s. The segment length is calculated as $N_s = (N/s)$. If s and N are incommensurate numbers, i.e., a reminder of the division of N by s is a non-zero number, a similar procedure of dividing the signal is repeated from the other end of the signal (i.e., from N to 1) so that entire signal is included in the analysis. Thus, the total number of segments of the signals are $2N_s$.

- **Step - 3:** Now, fit a suitable degree polynomial (by least-square fitting) to every segment of $2N_s$ of the signal separately. In the local detrending procedure, this calculated least-square fit of the segment is subtracted from the original segment (i.e., calculate the residue of the segment), and the variance of this subtracted signal is obtained as,
  $F^2(v, s) = \frac{1}{s}\sum_{i=1}^{s}\{Y[(v-1)s + i] - y_v(i)\}^2$ for segments, $v = 1, 2, \cdots, N_s$
  and,
  $F^2(v, s) = \frac{1}{s}\sum_{i=1}^{s}\{Y[N - (v - N_s)s + i] - y_v(i)\}^2$ for segments, $v = N_s + 1, \cdots, 2N_s$
  where, $y_v(i)$ is a polynomial fitted to a segment of the integrated signal. The choice of order (q) of the polynomial could be linear ($q = 1$), quadratic ($q = 2$), cubic ($q = 3$) or any higher order value ($q > 3$). As the original signal (x_k) is multifractal (involving different scales, slow as well as fast), different orders of detrending are required to remove the corresponding trends present in the signal.

- **Step - 4:** The q^{th}- order average of variance of all the segments is obtained as,

$$F_q(s) = \left\{ \frac{1}{2N_s} \sum_{v=1}^{2N_s} [F^2(v, s)]^{q/2} \right\}^{1/q} \tag{11}$$

  where, the value of q ranges from anywhere between positive and negative except zero. In the detrended fluctuation analysis of monofractal signals, the value of $q = 2$ (second order statistics) is sufficient, whereas, for the multifractal signal, different values of q are necessary to quantify small as well as large magnitude events of the signal. The negative values of q influence the small fluctuations, whereas, positive value of q influence the large fluctuations. When $q = 0$, the above mentioned formula of $F_q(s)$ is modified to a logarithmic form as given below,

$$F_{q=0}(s) = \exp\left\{ \frac{1}{2N_s} \sum_{v=1}^{2N_s} \ln[F^2(v, s)] \right\} \tag{12}$$

- **Step - 5:** The scaling behavior of fluctuations in the signal can be determined from the slope of the log-log plot of $F_q(s)$ versus s obtained for each value of q. The quantification over different segments of the sample sizes (s) is important in characterizing both fast changing fluctuations (changing over small segments) and the slow changing fluctuations (changing over large segments) of the multifractal signal. The plot $F_q(s)$ versus s shows a power law behavior, whose slope is called a Hurst exponent (H). The value of the Hurst exponent is always positive, indicating an increasing value of overall q-order statistics with the increase in size of the segment (s). Thus, the long range power law correlation of the signal is approximated as,

$$F_q(s) \sim s^{H(q)} \tag{13}$$

  where $H(q)$ is called a generalized Hurst exponent. The value of $H(q)$ is independent of q for a monofractal signal. On the other hand, the dependency of $H(q)$ over q becomes apparent for multifractal signals. The value of H ranges from 0 to 1 for noise like signals, and has values greater than 1 for random walk like signals. When H lies in between 0 to 0.5, the signal is called anti-persistent (correlated), and when H has values in between 0.5 to 1, the signal is called persistent (anti-correlated). When $H = 0$, the times series has independent (or zero memory) structures. White noise signals have H value close to 0.5, whereas, for monofractal or multifractal signals, the value of H ranges in between 0.5 to 1.

- **Step - 6:** *Multifractal Spectrum*
  As mentioned earlier, a generalized Hurst exponent $H(q)$ is used to represent the behavior of multifractal signals. However, in practice, it is only one of the possible types of scaling exponents used to characterize the

multifractal nature of real signals. In the typical literature of multifractal analysis, this $H(q)$ is converted into a q^{th} order mass exponent $\tau(q)$, then the relation between $H(q)$ and $\tau(q)$ is given by

$$\tau(q) = qH(q) - 1 \tag{14}$$

Using the value of $\tau(q)$, we can calculate the singularity spectrum $f(\alpha)$ of the multifractal signal using a Legendre transform as follows (68),

$$f(\alpha) = q\alpha - \tau(q) \tag{15}$$

where $\alpha = \frac{\partial \tau(q)}{\partial q}$ and is known a singularity strength or Hölder exponent.
Thus, the multifractal spectrum (also known as the Hölder spectrum) is a plot between $f(\alpha)$ and α. This plot provides the information about the varying nature of the fractal structure (scaling exponents) of the data.
The shape of the spectrum is like an inverted parabola, wherein, the difference between the maximum and minimum values of α is called the width of the spectrum, and is given by spectrum width $= \alpha_{max} - \alpha_{min}$. The width of the multifractal spectrum represents the deviation of the fractal structure of the signal from its average value (also known as the strength of the multifractality). The wider the width of the spectrum, the richer is the multifractal nature of the signal. For a monofractal signal, the shape of the spectrum converges to a single point for which $f(\alpha) = 1$.

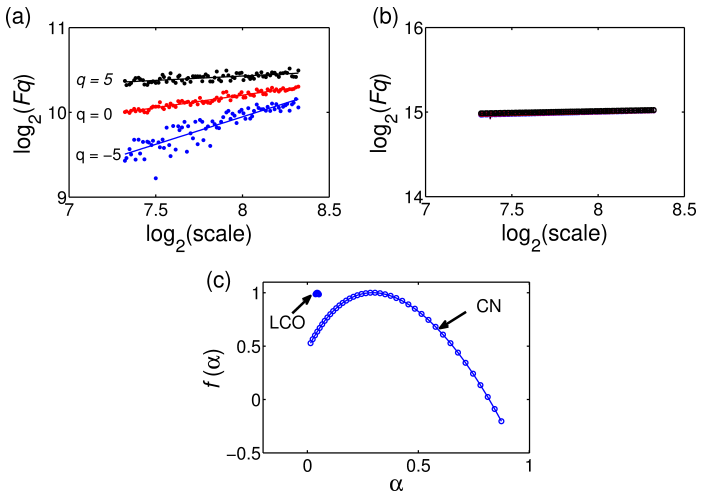

Figure 23

(a), (b) Logarithmic plots of overall root mean square value of the fluctuations versus different values of the scales, plotted for three values of orders (q): -5, 0, 5. (a) A combustion noise state and (b) a thermoacoustic instability (limit cycle) state. In (a), the slope of the plot $F_q(s)$ versus s is known as the generalized Hurst exponent. In the case of combustion noise (a), the value of the Hurst exponent decreases as the order of the fluctuations increases from negative to positive values. On the other hand, for the limit cycle state, all the curves of $F_q(s)$ versus s for different q collapse into a single curve, depicting the same value of Hurst exponent for all orders of the fluctuations. (c) The multifractal spectrum of the acoustic pressure signals, characterized using the plot of singularity spectrum $f(\alpha)$ with singularity strength α, obtained during the combustion noise (CN) state (shown in a) and during thermoacoustic instability (LCO) state (shown in b). The spectrum of combustion noise shows a wide variation, whereas that for the limit cycle collapses to a single point. Thus, the transition of the spectrum from a broad structure to a single point depicts the behavior of loss of multifractality of the acoustic pressure oscillations during the transition from combustion noise to thermoacoustic instability.

Nair and Sujith (43) used the MFDFA technique for characterizing the multifractality of the signals obtained from thermoacoustic systems. They observed that combustion noise is multifractal and has various scales corresponding to hydrodynamics, acoustics or combustion processes associated with it. This multifractal nature of the system dynamics is lost during the onset of thermoacoustic instability in the system. With the help of a measure of multifractality of the acoustic pressure signals, i.e., the Hurst exponent, they were able to forewarn the onset of thermoacoustic instabilities in turbulent combustion systems. Furthermore, Unni and Sujith (49) studied the blowout characteristics of a turbulent combustion system using MFDFA analysis. Examples and computer programmes for performing MFDFA can be found in ref. (67).

Acknowledgement

The help from Mr. Samadhan Ananda Pawar and Dr. Sirshendu Mondal is gratefully acknowledged.

LITERATURE CITED

1. A. V. Oppenheim, *Discrete-time signal processing*. Pearson Education India, 1999.
2. E. Lai, *Practical digital signal processing*. Newnes, 2003.
3. L. Tan and J. Jiang, *Fundamentals of analog and digital signal processing*. AuthorHouse, 2007.
4. S. Smith, *Digital signal processing: a practical guide for engineers and scientists*. Newnes, 2013.
5. C. E. Shannon, "Communication in the presence of noise," *Proceedings of the IRE*, vol. 37, no. 1, pp. 10–21, 1949.
6. R. C. Hilborn, *Chaos and nonlinear dynamics: an introduction for scientists and engineers*. Oxford University Press on Demand, 2000.
7. J. M. T. Thompson and H. B. Stewart, *Nonlinear dynamics and chaos*. John Wiley & Sons, 2002.
8. A. H. Nayfeh and B. Balachandran, *Applied nonlinear dynamics: analytical, computational and experimental methods*. John Wiley & Sons, 2008.
9. H. Poincaré, "Sur l'équilibre d'une masse fluide animée d'un mouvement de rotation," *Acta mathematica*, vol. 7, no. 1, pp. 259–380, 1885.
10. S. H. Strogatz, *Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering*. Westview press, 2014.
11. J. D. Crawford, "Introduction to bifurcation theory," *Reviews of Modern Physics*, vol. 63, no. 4, p. 991, 1991.
12. F. Cuesta, E. Ponce, and J. Aracil, "Local and global bifurcations in simple takagi-sugeno fuzzy systems," *IEEE Transactions on Fuzzy Systems*, vol. 9, no. 2, pp. 355–368, 2001.
13. M. Orlik, *Self-organization in electrochemical systems I: General principles of self-organization. Temporal instabilities*. Springer Science & Business Media, 2012.
14. N. Ananthkrishnan, S. Deo, and F. E. Culick, "Reduced-order modeling and dynamics of nonlinear acoustic waves in a combustion chamber," *Combustion science and technology*, vol. 177, no. 2, pp. 221–248, 2005.
15. P. Subramanian, S. Mariappan, R. Sujith, and P. Wahi, "Bifurcation analysis of thermoacoustic instability in a horizontal rijke tube," *International journal of spray and combustion dynamics*, vol. 2, no. 4, pp. 325–355, 2010.
16. R. Sujith, M. Juniper, and P. Schmid, "Non-normality and nonlinearity in thermoacoustic instabilities," *International Journal of Spray and Combustion Dynamics*, vol. 8, no. 2, pp. 119–146, 2016.
17. T. C. Lieuwen, "Experimental investigation of limit-cycle oscillations in an unstable gas turbine combustor," *Journal of Propulsion and Power*, vol. 18, no. 1, pp. 61–67, 2002.
18. S. Etikyala and R. Sujith, "Change of criticality in a prototypical thermoacoustic system," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 27, no. 2, p. 023106, 2017.
19. L. Kabiraj, A. Saurabh, P. Wahi, and R. Sujith, "Route to chaos for combustion instability in ducted laminar premixed flames," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 22, no. 2, p. 023129, 2012.
20. E. N. Lorenz, "Deterministic nonperiodic flow," *Journal of the atmospheric sciences*, vol. 20, no. 2, pp. 130–141, 1963.
21. Y. Pomeau and P. Manneville, "Intermittent transition to turbulence in dissipative dynamical systems," *Communications in Mathematical Physics*, vol. 74, no. 2, pp. 189–197, 1980.
22. J.-P. Eckmann and D. Ruelle, "Ergodic theory of chaos and strange attractors," *Reviews of modern physics*, vol. 57, no. 3, p. 617, 1985.
23. N. H. Packard, J. P. Crutchfield, J. D. Farmer, and R. S. Shaw, "Geometry from a time series," *Physical review letters*, vol. 45, no. 9, p. 712, 1980.
24. H. D. Abarbanel, R. Brown, J. J. Sidorowich, and L. S. Tsimring, "The analysis of observed chaotic data in physical systems," *Reviews of modern physics*, vol. 65, no. 4, p. 1331, 1993.
25. M. B. Kennel, R. Brown, and H. D. Abarbanel, "Determining embedding dimension for phase-space reconstruction using a geometrical construction," *Physical review A*, vol. 45, no. 6, p. 3403, 1992.
26. F. Takens *et al.*, "Detecting strange attractors in turbulence," *Lecture notes in mathematics*, vol. 898, no. 1, pp. 366–381, 1981.
27. S. A. Pawar, A. Seshadri, U. VR, and R. Sujith, "Thermoacoustic instability as mutual synchronization between the acoustic field of the confinement and turbulent reactive flow," *Journal of Fluid Mechanics*, pp. in–press, 2017.
28. H. D. Abarbanel and M. B. Kennel, "Local false nearest neighbors and dynamical dimensions from observed chaotic data," *Physical Review E*, vol. 47, no. 5, p. 3057, 1993.
29. H. Kantz and T. Schreiber, *Nonlinear time series analysis*, vol. 7. Cambridge university press, 2004.
30. H. G. Schuster and W. Just, *Deterministic chaos: an introduction*. John Wiley & Sons, 2006.
31. A. M. Fraser and H. L. Swinney, "Independent coordinates for strange attractors from mutual information," *Physical review A*, vol. 33, no. 2, p. 1134, 1986.
32. L. Kabiraj and R. Sujith, "Nonlinear self-excited thermoacoustic oscillations: intermittency and flame blowout," *Journal of Fluid Mechanics*, vol. 713, pp. 376–397, 2012.
33. L. Cao, "Practical method for determining the minimum embedding dimension of a scalar time series," *Physica D: Nonlinear Phenomena*, vol. 110, no. 1-2, pp. 43–50, 1997.
34. V. Nair, G. Thampi, S. Karuppusamy, S. Gopalan, and R. Sujith, "Loss of chaos in combustion noise as a precursor of impending combustion instability," *International journal of spray and combustion dynamics*, vol. 5, no. 4, pp. 273–290,

2013.
35. J. Tony, E. Gopalakrishnan, E. Sreelekha, and R. Sujith, "Detecting deterministic nature of pressure measurements from a turbulent combustor," *Physical Review E*, vol. 92, no. 6, p. 062902, 2015.
36. G. L. Baker and J. P. Gollub, *Chaotic dynamics: an introduction*. Cambridge University Press, 1996.
37. H. Poincaré, "Sur le problème des trois corps et les équations de la dynamique," *Acta mathematica*, vol. 13, no. 1, pp. A3–A270, 1890.
38. J.-P. Eckmann, S. O. Kamphorst, and D. Ruelle, "Recurrence plots of dynamical systems," *EPL (Europhysics Letters)*, vol. 4, no. 9, p. 973, 1987.
39. N. Marwan, M. C. Romano, M. Thiel, and J. Kurths, "Recurrence plots for the analysis of complex systems," *Physics reports*, vol. 438, no. 5, pp. 237–329, 2007.
40. S. Schinkel, O. Dimigen, and N. Marwan, "Selection of recurrence threshold for signal detection," *The European Physical Journal-Special Topics*, vol. 164, no. 1, pp. 45–53, 2008.
41. N. Marwan, "How to avoid potential pitfalls in recurrence plot based data analysis," *International Journal of Bifurcation and Chaos*, vol. 21, no. 04, pp. 1003–1017, 2011.
42. K. Klimaszewska and J. J. Żebrowski, "Detection of the type of intermittency using characteristic patterns in recurrence plots," *Physical review E*, vol. 80, no. 2, p. 026214, 2009.
43. V. Nair, G. Thampi, and R. Sujith, "Intermittency route to thermoacoustic instability in turbulent combustors," *Journal of Fluid Mechanics*, vol. 756, pp. 470–487, 2014.
44. H. Gotoda, Y. Shinoda, M. Kobayashi, Y. Okuno, and S. Tachibana, "Detection and control of combustion instability based on the concept of dynamical system theory," *Physical Review E*, vol. 89, no. 2, p. 022910, 2014.
45. V. Nair and R. Sujith, "Intermittency as a transition state in combustor dynamics: An explanation for flame dynamics near lean blowout," *Combustion Science and Technology*, vol. 187, no. 11, pp. 1821–1835, 2015.
46. L. Kabiraj, H. Nawroth, A. Saurabh, and C. O. Paschereit, "Recurrence analysis of combustion generated noise," in *52nd Aerospace Sciences Meeting*, pp. 2014–0623, 2014.
47. L. Christodoulou, L. Kabiraj, A. Saurabh, and N. Karimi, "Characterizing the signature of flame flashback precursor through recurrence analysis," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 26, no. 1, p. 013110, 2016.
48. U. Sen, T. Gangopadhyay, C. Bhattacharya, A. Misra, S. Karmakar, P. Sengupta, A. Mukhopadhyay, and S. Sen, "Investigation of ducted inverse nonpremixed flame using dynamic systems approach," in *ASME Turbo Expo 2016: Turbomachinery Technical Conference and Exposition*, pp. V04BT04A059–V04BT04A059, American Society of Mechanical Engineers, 2016.
49. V. R. Unni and R. Sujith, "Multifractal characteristics of combustor dynamics close to lean blowout," *Journal of Fluid Mechanics*, vol. 784, pp. 30–50, 2015.
50. S. A. Pawar, R. Vishnu, M. Vadivukkarasan, M. Panchagnula, and R. Sujith, "Intermittency route to combustion instability in a laboratory spray combustor," *Journal of Engineering for Gas Turbines and Power*, vol. 138, no. 4, p. 041505, 2016.
51. Z.-K. Gao, M. Small, and J. Kurths, "Complex network analysis of time series," *EPL (Europhysics Letters)*, vol. 116, no. 5, p. 50001, 2017.
52. R. V. Donner, Y. Zou, J. F. Donges, N. Marwan, and J. Kurths, "Recurrence networksa novel paradigm for nonlinear time series analysis," *New Journal of Physics*, vol. 12, no. 3, p. 033025, 2010.
53. L. Lacasa, B. Luque, F. Ballesteros, J. Luque, and J. C. Nuno, "From time series to complex networks: The visibility graph," *Proceedings of the National Academy of Sciences*, vol. 105, no. 13, pp. 4972–4975, 2008.
54. S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang, "Complex networks: Structure and dynamics," *Physics reports*, vol. 424, no. 4, pp. 175–308, 2006.
55. S. Bornholdt and H. G. Schuster, *Handbook of graphs and networks: from the genome to the internet*. John Wiley & Sons, 2006.
56. X. F. Wang and G. Chen, "Complex networks: small-world, scale-free and beyond," *IEEE circuits and systems magazine*, vol. 3, no. 1, pp. 6–20, 2003.
57. D. J. Watts and S. H. Strogatz, "Collective dynamics of small-worldnetworks," *nature*, vol. 393, no. 6684, pp. 440–442, 1998.
58. P. Erdos and A. Rényi, "On the evolution of random graphs," *Publ. Math. Inst. Hung. Acad. Sci*, vol. 5, no. 1, pp. 17–60, 1960.
59. A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *science*, vol. 286, no. 5439, pp. 509–512, 1999.
60. R. Albert and A.-L. Barabási, "Statistical mechanics of complex networks," *Reviews of modern physics*, vol. 74, no. 1, p. 47, 2002.
61. J. Travers and S. Milgram, "The small world problem," *Phychology Today*, vol. 1, pp. 61–67, 1967.
62. M. Murugesan and R. Sujith, "Combustion noise is scale-free: transition from scale-free to order at the onset of thermoacoustic instability," *Journal of Fluid Mechanics*, vol. 772, pp. 225–245, 2015.
63. Y. Okuno, M. Small, and H. Gotoda, "Dynamics of self-excited thermoacoustic instability in a combustion system: Pseudo-periodic and high-dimensional nature," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 25, no. 4, p. 043107, 2015.
64. J. W. Kantelhardt, E. Koscielny-Bunde, H. H. Rego, S. Havlin, and A. Bunde, "Detecting long-range correlations with detrended fluctuation analysis," *Physica A: Statistical Mechanics and its Applications*, vol. 295, no. 3, pp. 441–454,

2001.
65. H. E. Hurst, "Long-term storage capacity of reservoirs," *Transactions of the American Society of Civil Engineers*, vol. 116, pp. 770–808, 1951.
66. J. W. Kantelhardt, S. A. Zschiegner, E. Koscielny-Bunde, S. Havlin, A. Bunde, and H. E. Stanley, "Multifractal detrended fluctuation analysis of nonstationary time series," *Physica A: Statistical Mechanics and its Applications*, vol. 316, no. 1, pp. 87–114, 2002.
67. E. A. Ihlen, "Introduction to multifractal detrended fluctuation analysis in matlab," *Frontiers in physiology*, vol. 3, 2012.
68. A. Chhabra and R. V. Jensen, "Direct determination of the f (α) singularity spectrum," *Physical Review Letters*, vol. 62, no. 12, p. 1327, 1989.