Linear & Nonlinear Systems

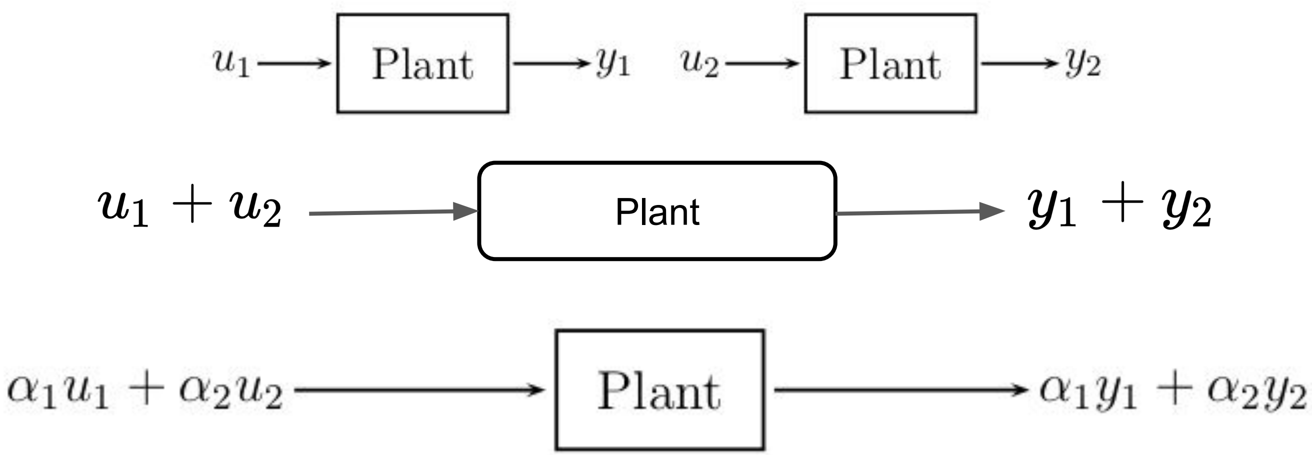
Lecture 2

Outline

- Linear Systems
- Impulse Response of Linear Systems
- LTI response to sinusoidal Inputs
- Analyzing LTI system behaviour
- Nonlinear System behaviour

Linear Systems

- A linear system follows the principle of ***superposition***.
- Superposition = ***Homogeneity*** + ***Additivity***
- Homogeneity: If input signal strength is increased (say doubled), the output response also increases in the same proportion.
- Additivity:



Example 1: Linear System

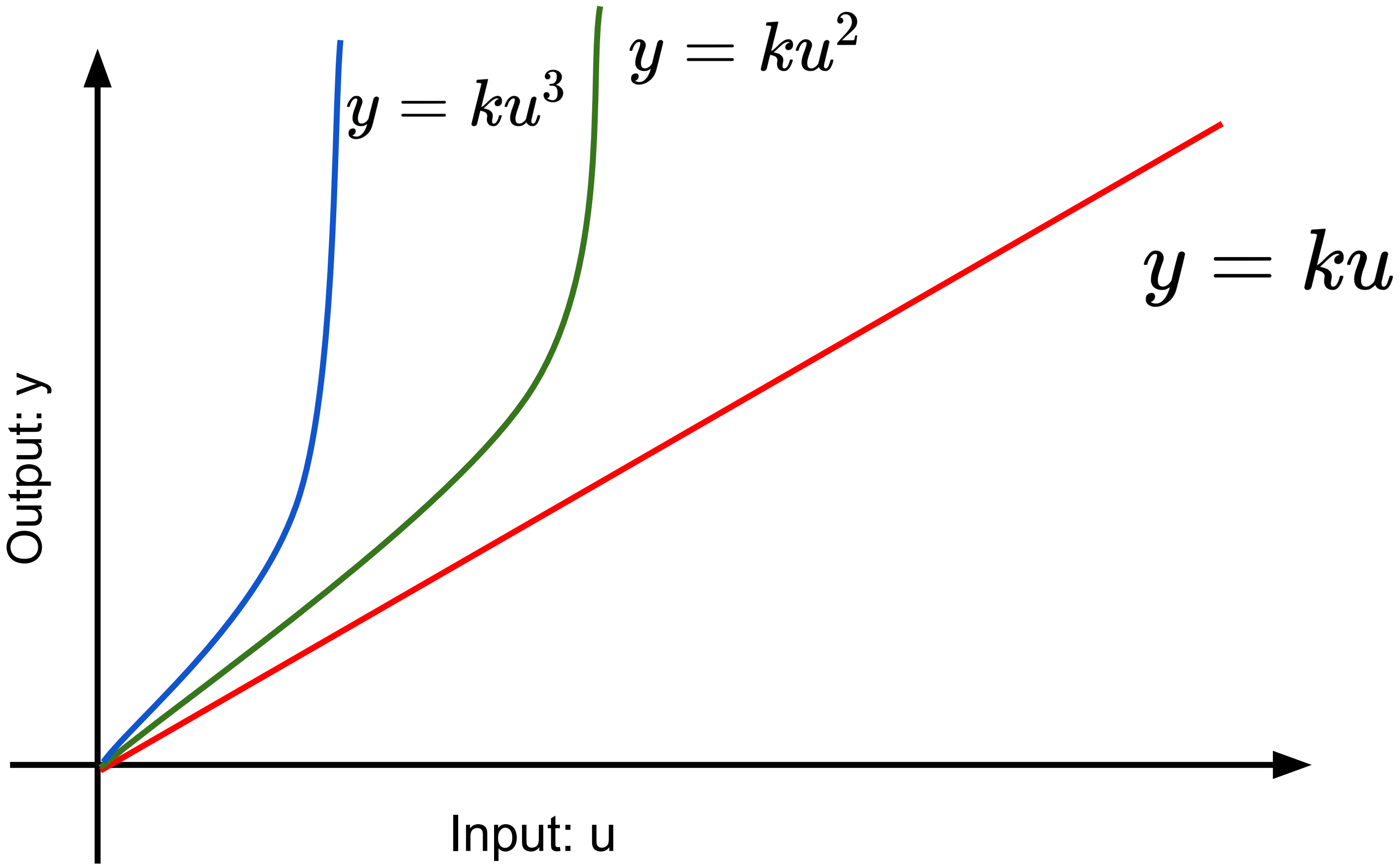
$$y = ku \qquad y_1 = ku_1;\; y_2 = ku_2$$

$$\alpha_1 u_1 + \alpha_2 u_2 \rightarrow \alpha_1 y_1 + \alpha_2 y_2$$

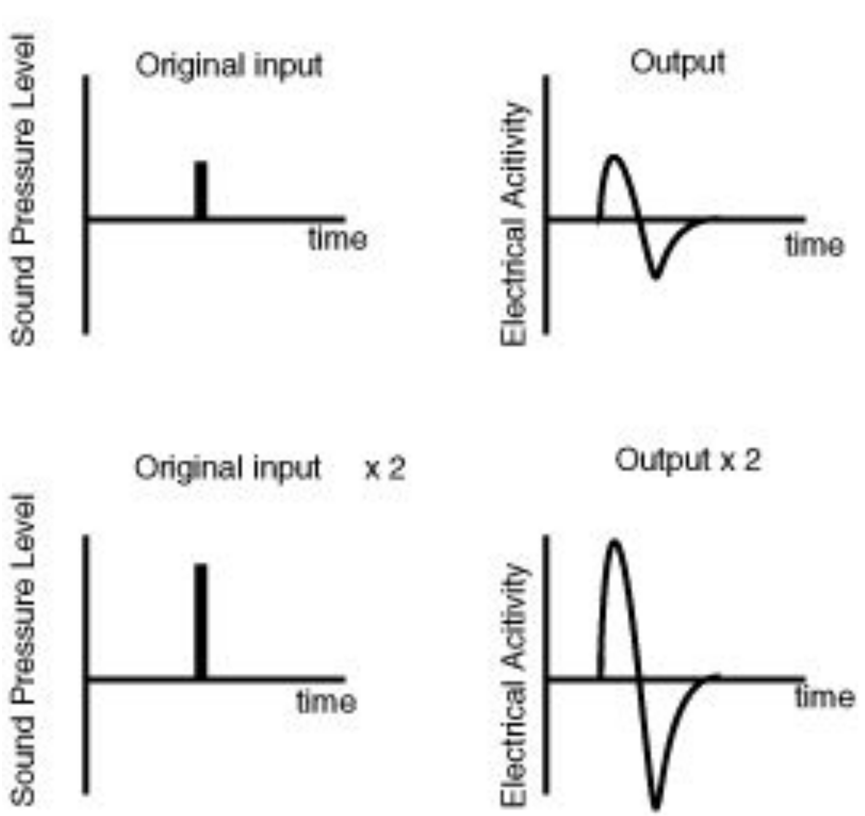
Example 2: Non-linear System

$$y = ku^2;\; y_1 = ku_1^2;\; y_2 = ku_2^2$$

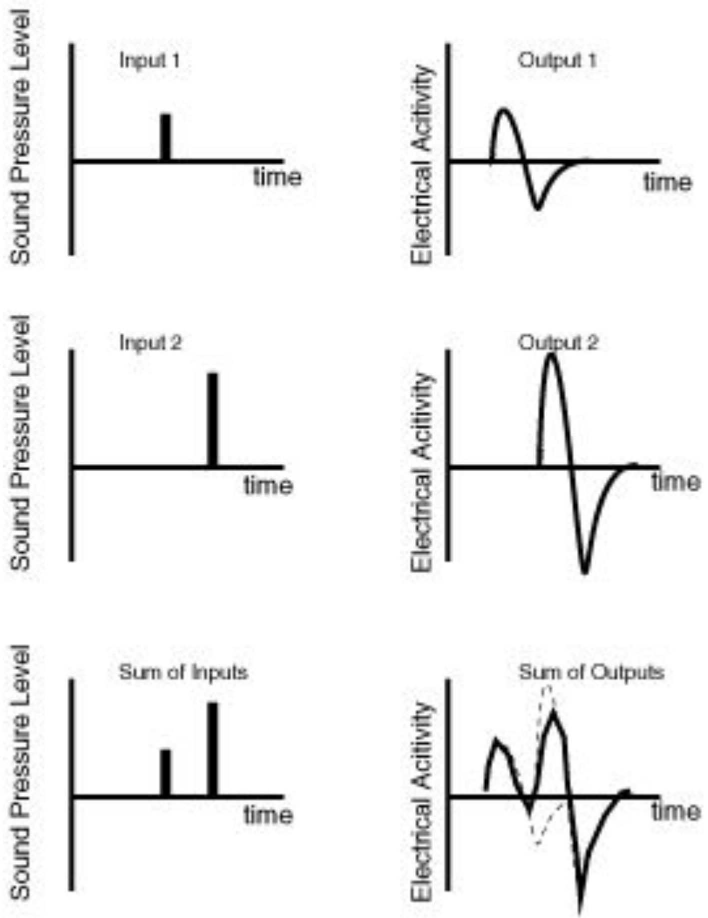
$$\alpha_1 u_1 + \alpha_2 u_2 \rightarrow \alpha_1^2 y_1 + \alpha_2^2 y_2$$


$y = ku^3$
$y = ku^2$
$y = ku$
Output: y
Input: u

Scalar Rule

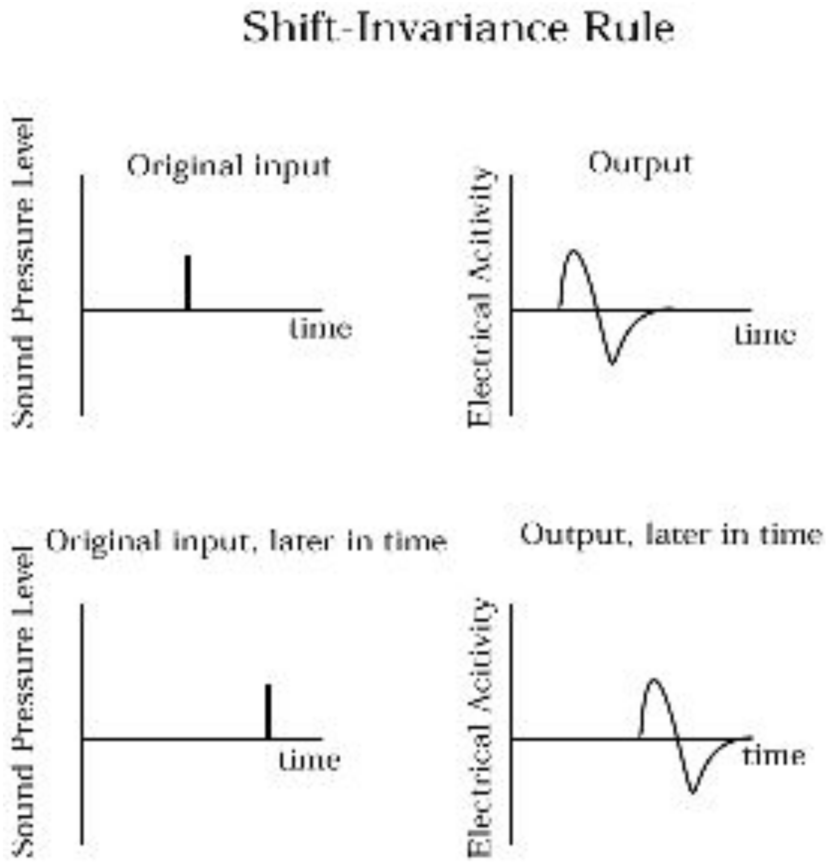


Additivity



Linear Time Invariant (LTI) Systems

- They are also called shift-invariant systems.
- If the input is shifted in time, the output also get shifted in time by exact amount.

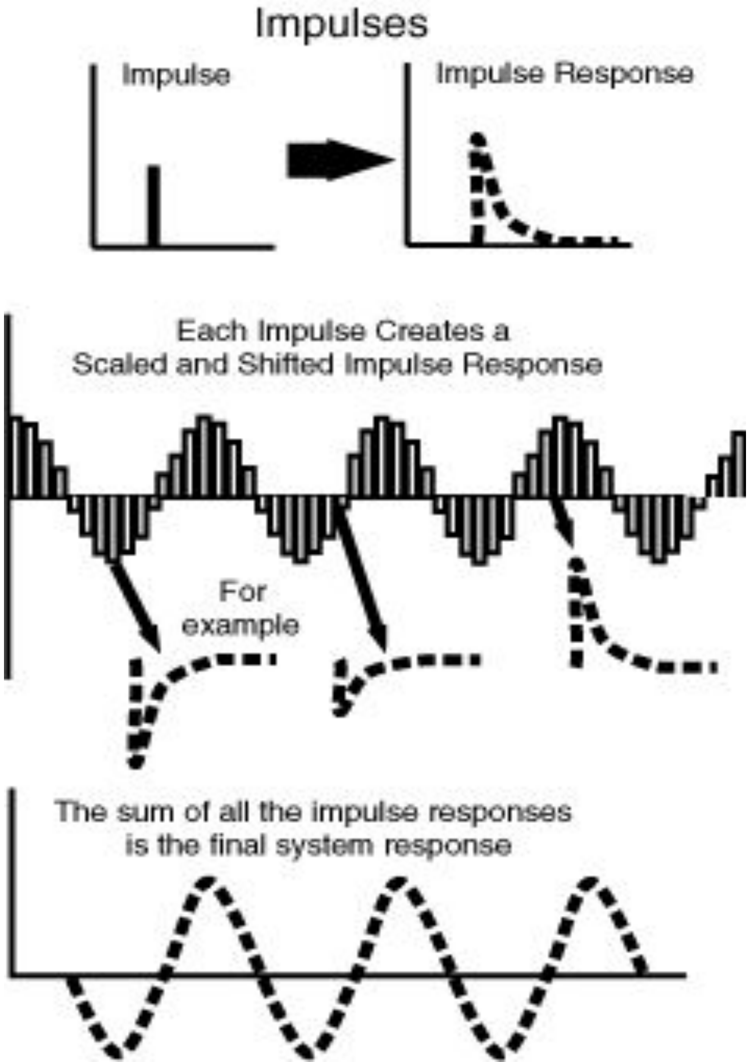


Impulse Response

- Superposition and Shift-invariance allows us to evaluate the response to an arbitrary input signal as a combination of various scaled and shifted impulse responses.

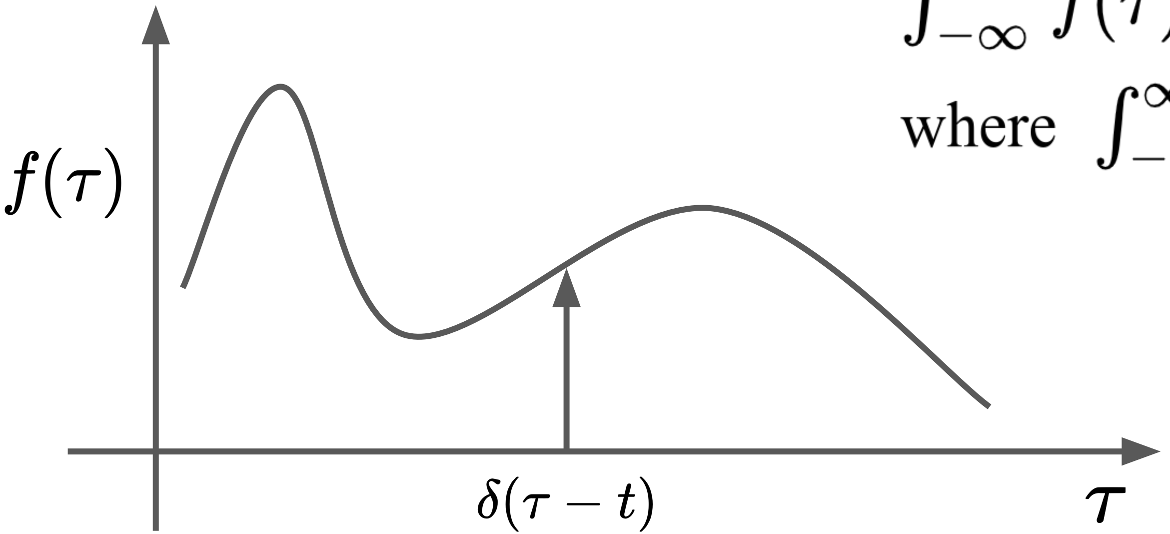
- Hence, in order to study the behaviour of an LTI system, we only need to know its ***impulse Response.***

- Impulse Response is the response of a system to an impulse input.



Definition of an Impulse

- A very intense force (or input) applied for a very short duration. Paul Dirac provided the following mathematical definition for an impulse function:

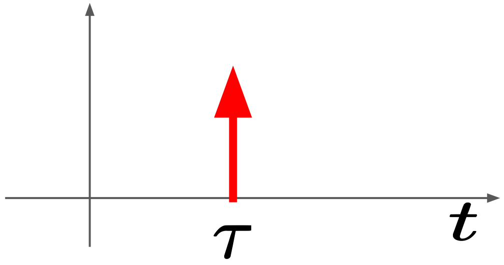


$$\int_{-\infty}^{\infty} f(\tau)\delta(\tau - t)d\tau = f(t)$$

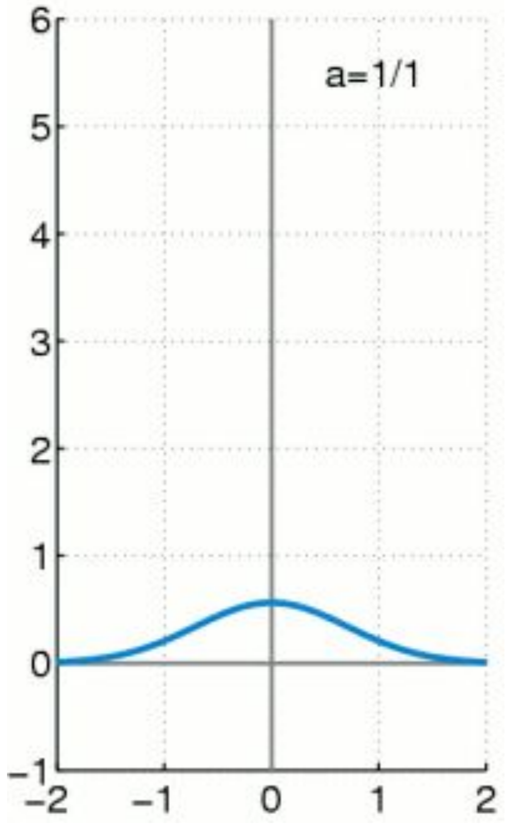
$$\text{where } \int_{-\infty}^{\infty} \delta(\tau - t)d\tau = 1$$

$$\delta(t) = \begin{cases} \infty \text{ if } t = 0 \\ 0 \text{ if } t \neq 0 \end{cases}$$

$$\delta(t-\tau) = \begin{cases} \infty \text{ if } t = \tau \\ 0 \text{ if } t \neq \tau \end{cases}$$



$$\delta_a(x) = \frac{1}{|a|\sqrt{\pi}} e^{-(x/a)^2} \text{ as } a \rightarrow 0$$

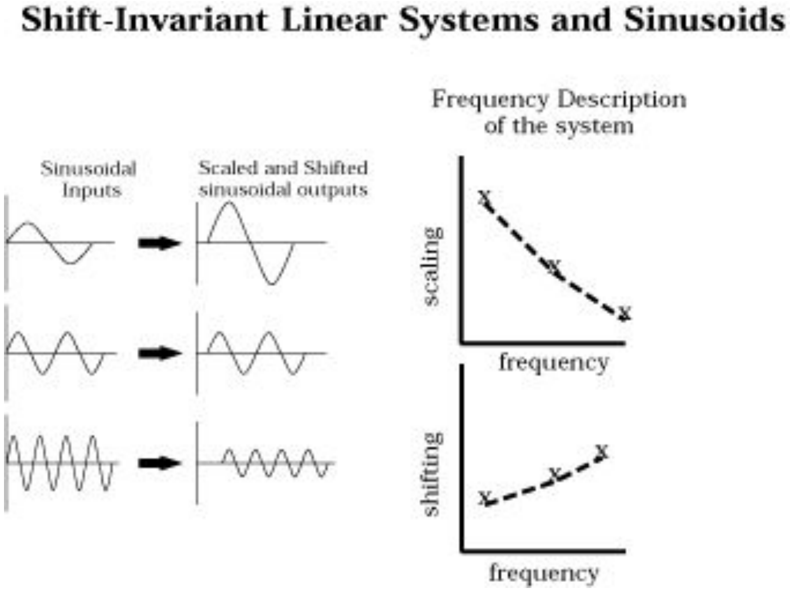


LTI Response to Sinusoidal Inputs

$$u(t) = A\sin(\omega t)$$

$$y(t) = B\sin(\omega t + \phi)$$

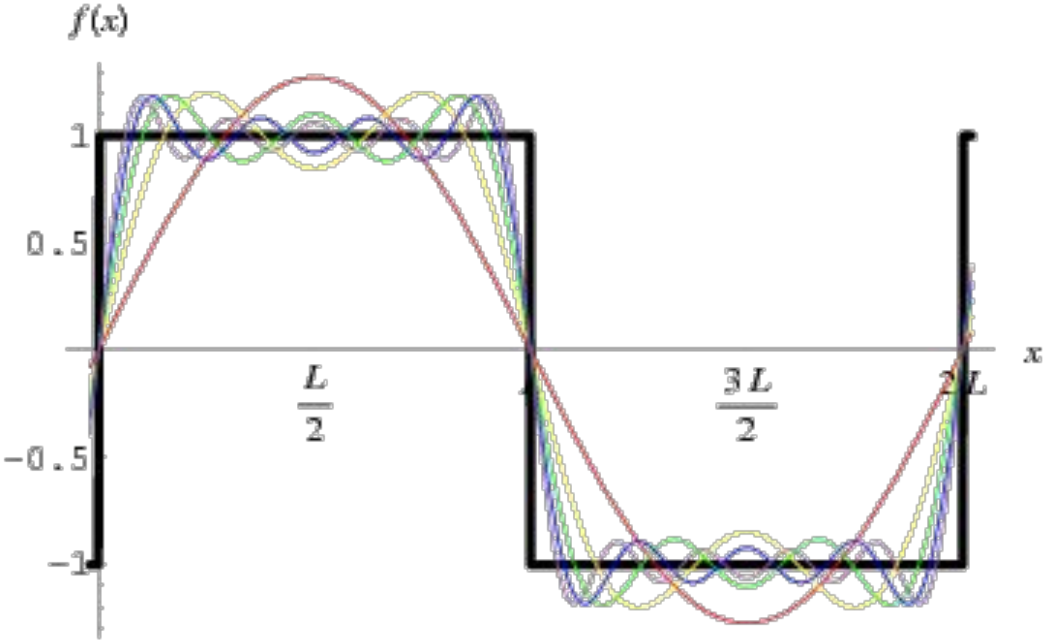
For sinusoidal input, the output of an LTI system is also sinusoidal with same frequency with a different scale (amplitude) and scale shift.



- Any periodic signal can be written as a sum of a series of shifted and scaled sinusoids at different frequencies - Known as fourier series:

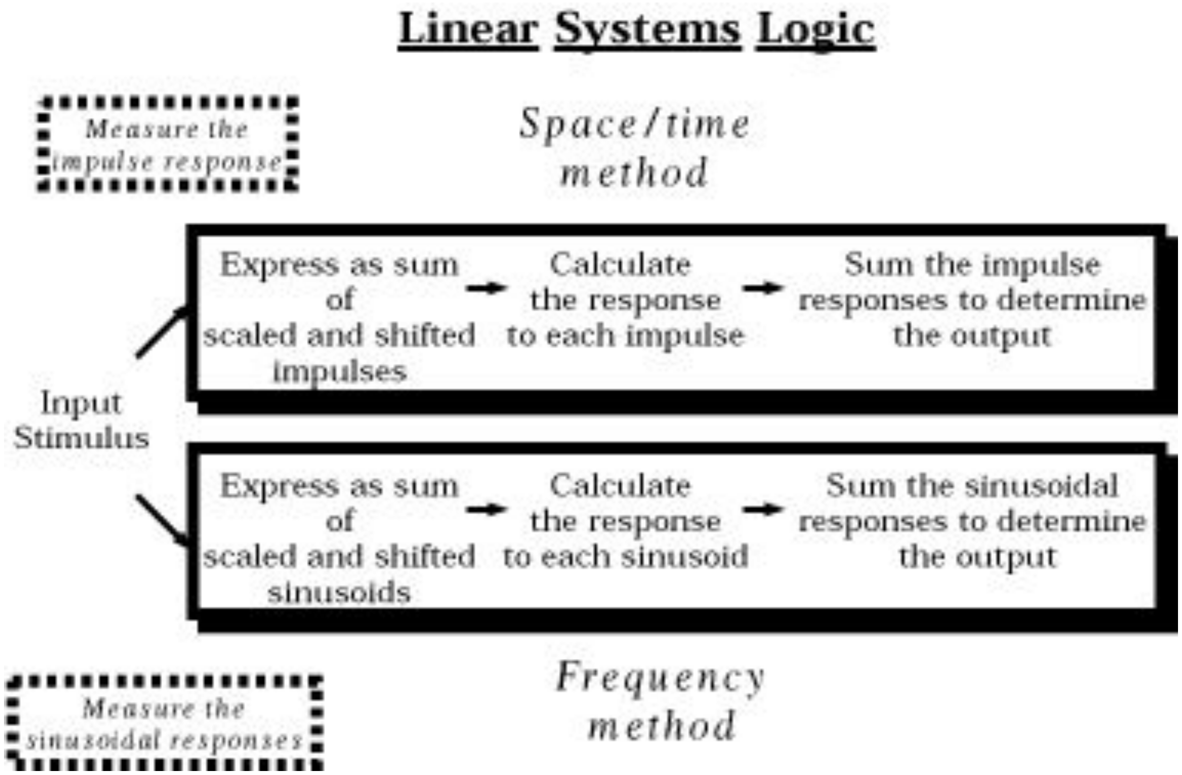
$$s(t) = A_0 + \sum_{i=1}^{\infty} A_i sin(\omega_i t + \phi_i)$$

- Hence, Superposition Principle allows finding LTI response to any periodic signal as a sum of responses to different sinusoidal signals.



Fourier Series approximation of a Square Wave

Analyzing Behaviour of Linear Systems



Nonlinear Systems

- A nonlinear dynamical system is typically represented as:

$$\dot{\mathbf{x}} = f(t, \mathbf{x}, u)$$

- Do not follow the superposition principle.
- May have multiple **Equilibrium** points.
- May exhibit phenomena like Limit Cycles, bifurcation and Chaos etc.

Equilibrium Point

- For a general system

$$\dot{x} = f(x, u)$$

The equilibrium points are those where

$$\dot{x} = 0 \Rightarrow f(x, u) = 0$$

- Sometimes, the nonlinear systems are ***linearized*** about their equilibrium point to convert them into Linear systems which are easier to analyze.

Linearization using Taylor's Series Expansion

If we perturb the system around its equilibrium point (x_e, u_e), the system dynamics can be expressed by a Taylor

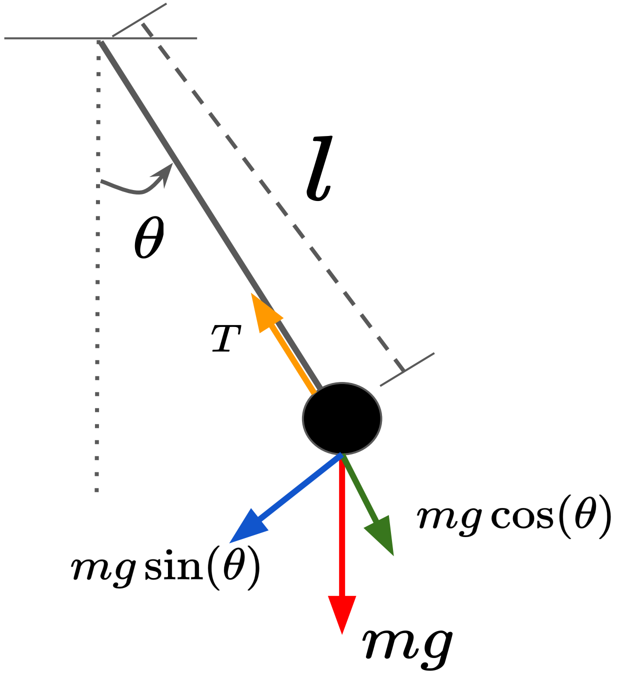
$$\begin{aligned}
\frac{dx}{dt} &= f(x_e + \Delta x, u_e + \Delta u) \\
\frac{d(x_e + \Delta x)}{dt} &= f(x_e, u_e) + \frac{\partial f}{\partial x}\Big|_{x_e, u_e} \Delta x + \frac{\partial f}{\partial u}\Big|_{x_e, u_e} \Delta u + \\
&\quad \frac{\partial^2 f}{\partial x^2}\Big|_{x_e, u_e} (\Delta x)^2 + \frac{\partial^2 f}{\partial u^2}\Big|_{x_e, u_e} (\Delta u)^2 + \ldots \\
\frac{d\Delta x}{dt} &= A\Delta x + B\Delta u
\end{aligned}$$

Only first order terms are retained in the Taylor series expansion. Hence, the linear system approximation around the equilibrium point is given by

$$\dot{\tilde{x}} = A\tilde{x} + B\tilde{u}$$ With $$A = \frac{\partial f}{\partial x}\Big|_{x_e, u_e} \qquad B = \frac{\partial f}{\partial u}\Big|_{x_e, u_e}$$

A Simple Pendulum

$$ml^2\ddot{\theta} + mgl\sin(\theta) + k\dot{\theta} = 0$$



$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = -\frac{g}{l}\sin(x_1) - \frac{k}{ml^2}x_2$$

Linearized model around origin is given by:

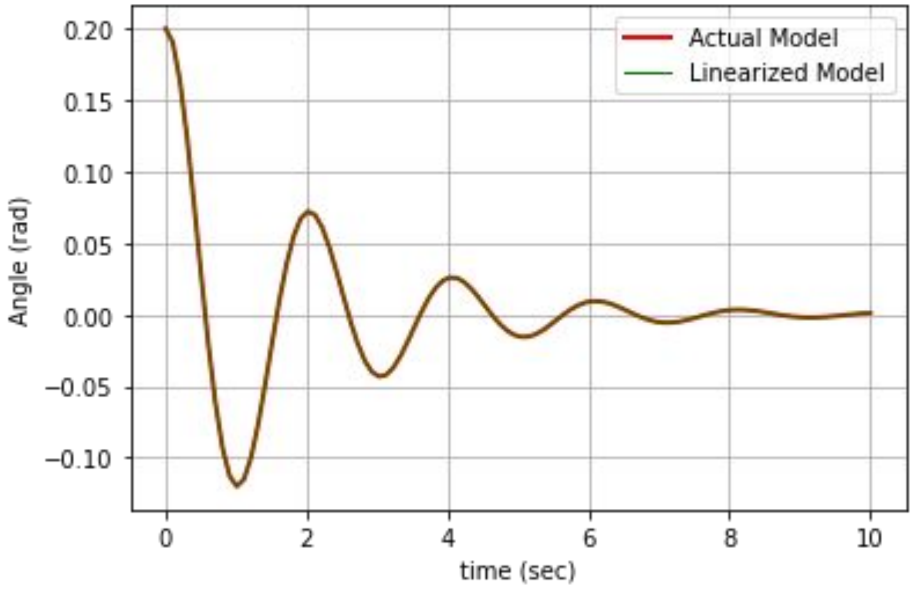
$$\dot{x} = Ax$$

$$A = \begin{bmatrix} 0 & 1 \\ -\frac{g}{l} & -\frac{k}{ml^2} \end{bmatrix}$$

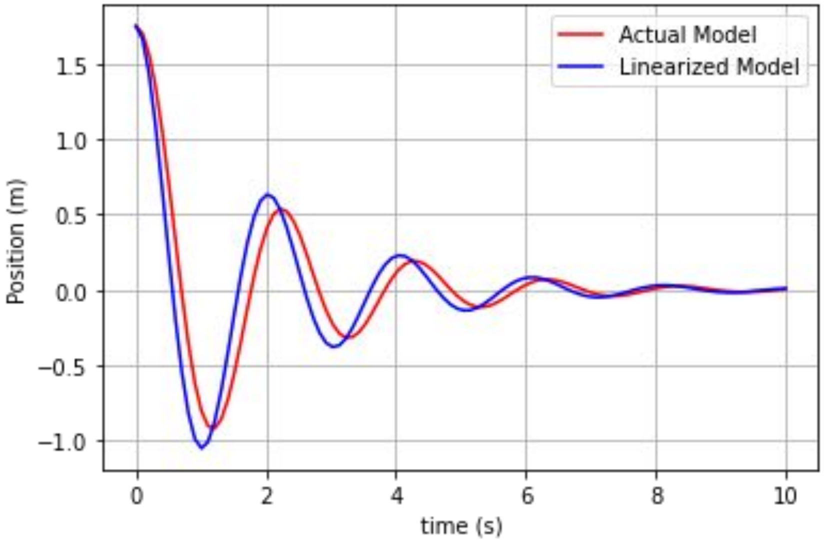
$$A = \begin{bmatrix} \frac{df_1}{dx_1} & \frac{df_1}{dx_2} \\ \frac{df_2}{dx_1} & \frac{df_2}{dx_2} \end{bmatrix} \text{ where } \begin{aligned} f_1 &= x_2 \\ f_2 &= -\frac{g}{l}\sin(x_1) - \frac{k}{ml^2}x_2 \end{aligned}$$

```python
import numpy as np
from scipy.integrate import odeint
import matplotlib.pyplot as plt
import math

# function that returns dy/dt
def non_linear_model(x,t):
    m = 0.5
    k = 0.5
    g = 9.8
    l = 1
    xdot = x[1]
    xddot = -(g/l) * math.sin(x[0]) - (k/(m*l*l)) * x[1]

    dxdt = [xdot, xddot]
    return dxdt

# function that returns dy/dt
def linear_model(x,t):
    m = 0.5
    k = 0.5
    g = 9.8
    l = 1
    xdot = x[1]
    xddot = -(g/l) * x[0] - k/(m*l*l) * x[1]

    dxdt = [xdot, xddot]
    return dxdt
```

```python
# initial condition
z0 = [1.75, 0] # try different values between 0 and 2.0

# time points
t = np.linspace(0,10, 100)

# solve ODE
z1 = odeint(non_linear_model,z0,t)
z2 = odeint(linear_model,z0,t)

print('shape of z:', np.shape(z1))

# plot results
plt.plot(t,z1[:,0], 'r-', label='Actual Model')
plt.plot(t, z2[:,0], 'b-', label='Linearized Model')
plt.xlabel('time (s)')
plt.ylabel('Position (m)')
plt.legend(loc='best')
plt.grid()
plt.show()
```



$$\theta(0) = 0.2$$



$$\theta(0) = 1.75$$

$$M = 0.5, g = 9.8, l = 1, k = 0.5$$

Multiple Equilibrium Points

Consider the following nonlinear equation:

$$\dot{x} = -x + x^2$$

Its equilibrium points are given by

$$\dot{x} = 0 \Rightarrow x(x-1) = 0$$

$$x = 0, \text{ and } x = 1$$

Linearized system around origin:

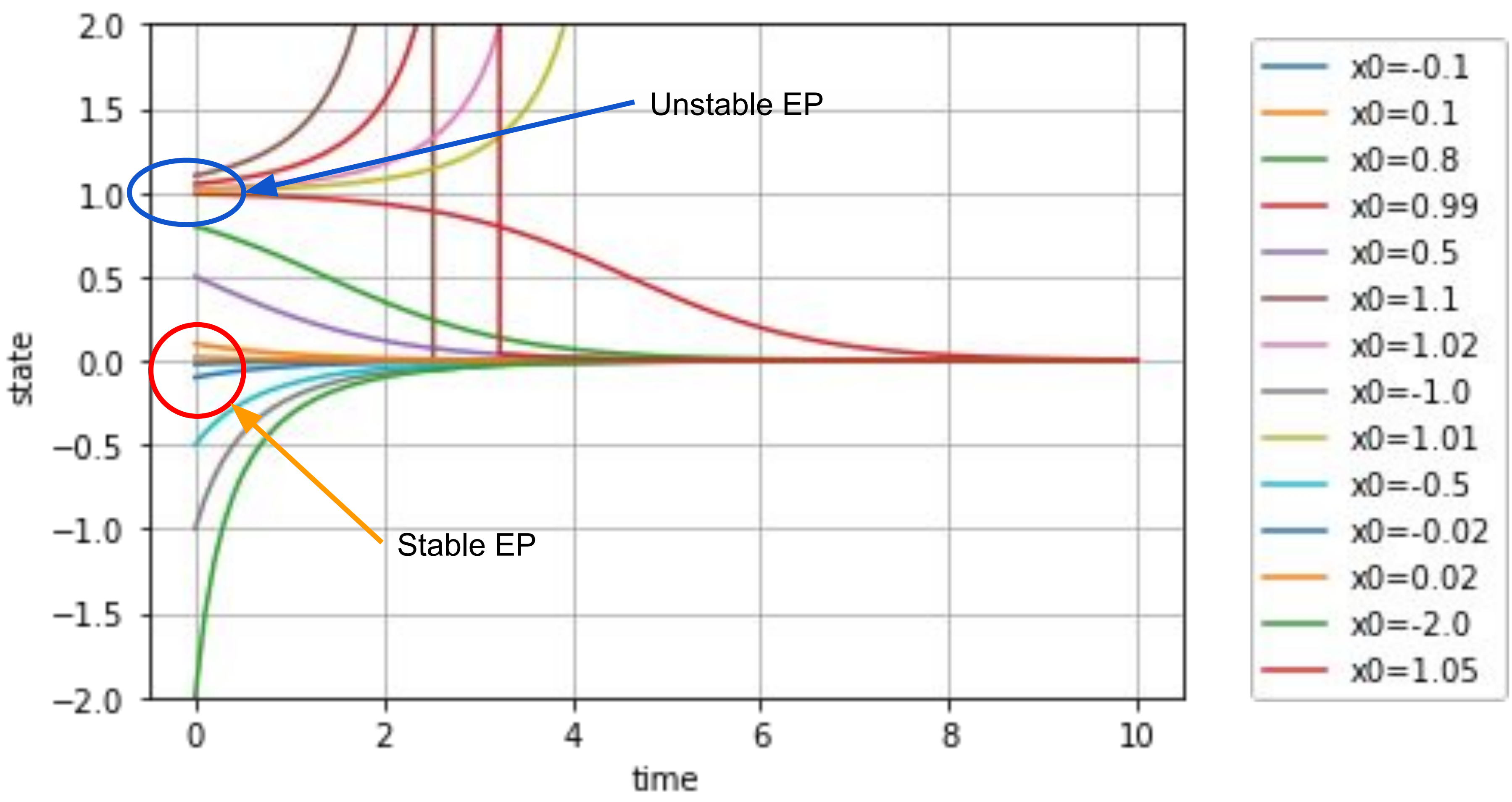
$$\dot{x} = \left.\frac{\partial f}{\partial x}\right|_{x=0} x$$

$$\dot{x} = -x$$

Solution of the linear system

$$x = x_0 e^{-t}$$

```python
import numpy as np
from scipy.integrate import odeint
import matplotlib.pyplot as plt

# function that returns dz/dt
def model(x,t):
    xdot = -x + x*x
    return xdot

# initial condition
x0 = [0.5, 0.3, 0.2, 1.0, -0.5, -1.0, -2.0, 0.99, 1.01, 1.2, 1.03, -0.2, -0.3]

# number of time points
n = 1000

fig = plt.figure()

# time points
t = np.linspace(0,15,n)

for i in range(len(x0)):
    x00 = x0[i]
    x = odeint(model, x00, t)
    plt.plot(t, x, '-', label='x0={}'.format(x0[i]))
    plt.ylim((-2,2))

plt.grid()
plt.xlabel('$time (sec)$')
plt.ylabel('$x$')
plt.legend(bbox_to_anchor=(1.05, 1))
plt.show()
```



Unstable EP
Stable EP
state
time
x0=-0.1
x0=0.1
x0=0.8
x0=0.99
x0=0.5
x0=1.1
x0=1.02
x0=-1.0
x0=1.01
x0=-0.5
x0=-0.02
x0=0.02
x0=-2.0
x0=1.05

Lorenz Attractor

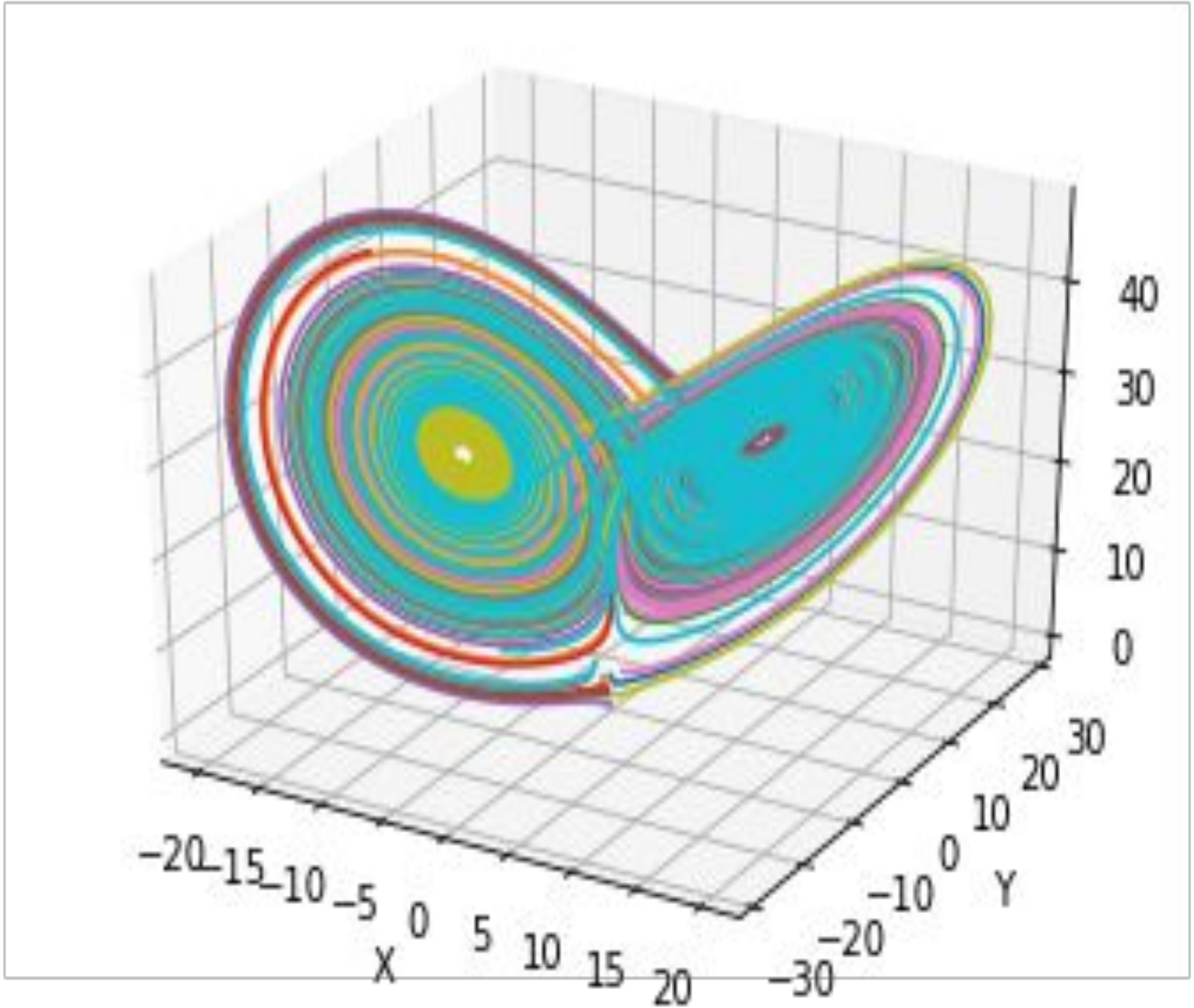
- It exhibits Chaotic Behaviour
- It does not retrace the same path twice or intersect other path.
- It shows randomness and unpredictability and yet strange kind of order.
- It never reaches a steady state or form Limit Cycles.
- It is an example of how a deterministic system can lead to unpredictable behaviour in the absence of the perfect knowledge of initial conditions.
- This is also known as the “butterfly effect” - flapping of a butterfly wing in Brazil can set off a tornado in Texas.

$$\begin{aligned} \dot{x} &= \sigma(y - x) \\ \dot{y} &= \rho x - y - xz \\ \dot{z} &= -\beta z + xy \end{aligned}$$

$$\sigma = 10$$

$$\rho = 28$$

$$\beta = 8/3$$




```
import numpy as np
from scipy.integrate import odeint
import matplotlib.pyplot as plt

# function that returns dz/dt
def model(Z,t):
    sigma = 10.0
    rho = 28.0
    beta = 8.0/3.0
    x = Z[0]
    y = Z[1]
    z = Z[2]
    dxdt = sigma * (y - x)
    dydt = rho * x - y - x * z
    dzdt = x * y - beta * z
    dz = [dxdt, dydt, dzdt]
    return dz

# initial condition
z0 = [1, 1, 1]

# number of time points
n = 10000

# time points
t = np.linspace(0,40,n)
# solve ODE
Z = odeint(model, z0, t)

fig = plt.figure()
ax = plt.axes(projection='3d')
ax.plot3D(Z[:,0], Z[:,1], Z[:,2], 'r-')
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
```

Vander-Pol Oscillator

- System Dynamics is given by the following equation:

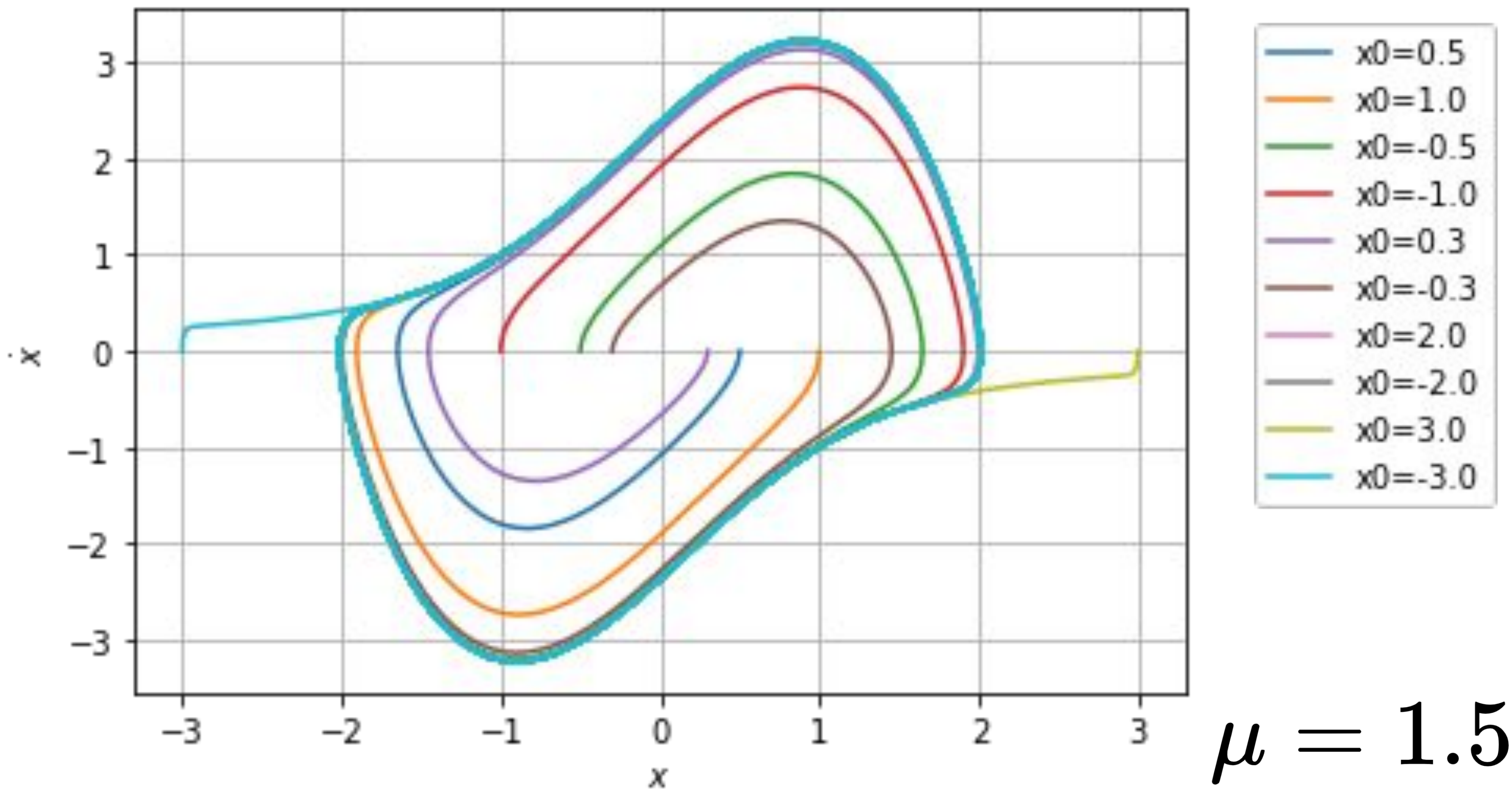
$$\ddot{x} + \mu(x^2 - 1)\dot{x} + x = 0$$

- It exhibits Limit Cycles - which are self-sustained oscillations even in the absence of external driving force.

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = -x_1 + \mu(1 - x_1^2)x_2$$

```
import numpy as np
from scipy.integrate import odeint
import matplotlib.pyplot as plt

# function that returns dz/dt
def model(z,t):
    mu = 1.5
    x = z[0]
    xdot = z[1]
    xddot = -x + mu * (1 - x*x)*xdot
    dz = [xdot, xddot]
    return dz

# initial condition
z00 = [0.5, 1.0, -0.5, -1.0, 0.3, -0.3, 2.0, -2.0, 3.0, -3.0]

# number of time points
n = 10000

fig = plt.figure()
# time points
t = np.linspace(0,40,n)

for i in range(len(z00)):
    z0 = [z00[i], 0]
    z = odeint(model, z0, t)
    plt.plot(z[:,0],z[:,1], label='x0={}'.format(z00[i]))

plt.grid()
plt.xlabel('$x$')
plt.ylabel('$\dot{x}$')
plt.legend(bbox_to_anchor=(1.05, 1))
plt.show()
```



$\mu = 1.5$

Summary

- In this module, we talked about the properties of linear and nonlinear systems.
- Linear systems follow the principle of superposition.
- LTI are shift-invariant.
- LTI systems behaviours can be analyzed by studying it impulse response and response to sinusoidal signals.
- Nonlinear systems do not follow the principle of superposition and are complex to analyze.
- Nonlinear systems can be linearized by using first order approximation of Taylor Series expansion.
- We saw several examples of nonlinear systems.