Modal Analysis for LTI Systems

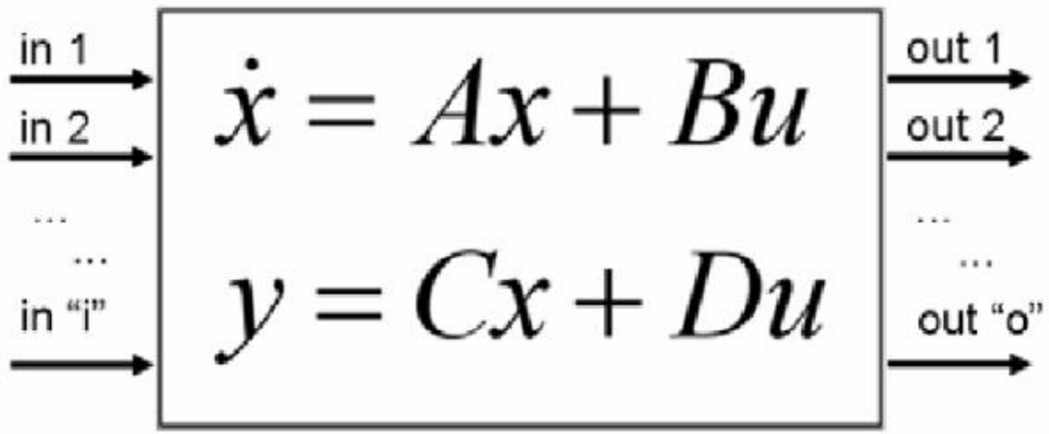
Week 8 – Advanced Topic 3

Definition

- To understand the behavior of dynamical systems by analyzing their modes.
- Modes are the natural ways in which the system responds to inputs or initial conditions.
- This analysis provides insights into how the system evolves over time.
- Gives idea of the stability, transient behavior, and resonance characteristics.
- LTIs have parameters that are not varying over time, i.e. constant.

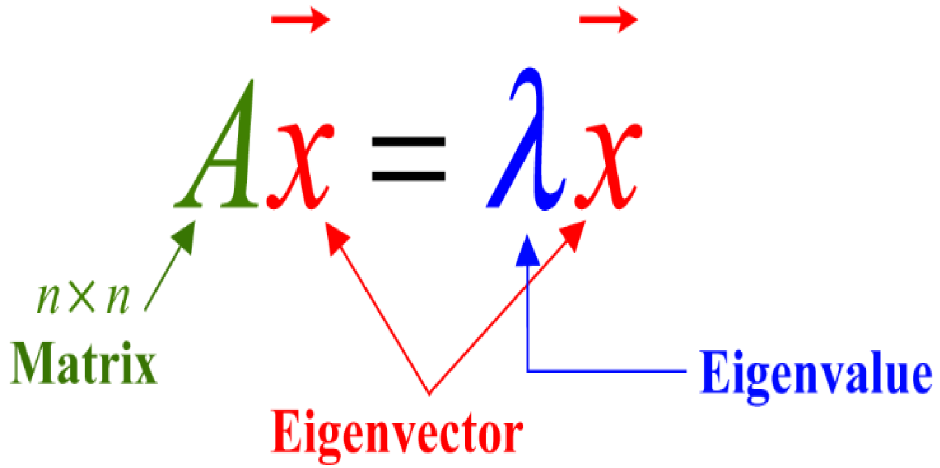
Steps

- LTI systems are often represented by **state-space equations** or **transfer functions**.
- A is the system matrix.
- B and C are input and output matrices.
- D is a direct feedthrough matrix.



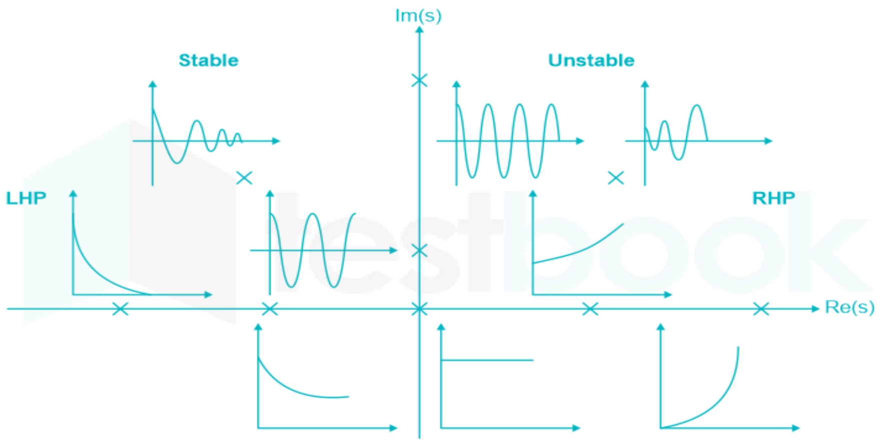
System Decomposition

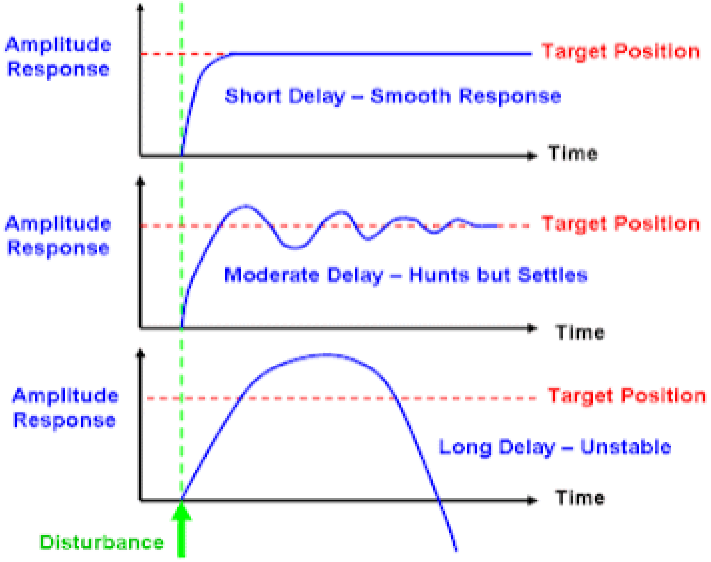
- Diagonalize the system matrix A.
- Find the **eigenvalues** and **eigenvectors** of the matrix.
- Eigenvalues (λ) represent system's natural frequencies (poles).
- Eigenvectors describe mode shapes or system response directions.



Mode interpretation

- The **eigenvalues** determine the stability and transient response of the system.
- If all eigenvalues have negative real parts, the system is stable.
- If any eigenvalue has a positive real part, the system is unstable.
- Complex eigenvalues correspond to oscillatory modes, where the real part represents damping and the imaginary part represents the oscillation frequency.
- The **eigenvectors** describe the shape of the mode and how the different states of the system are coupled in each mode.



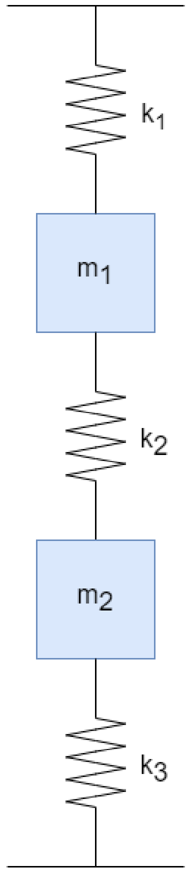


Modal Analysis of a spring mass system

- General equation
  - $m\ddot{u}(t) + C\dot{u}(t) + ku(t) = f(t)$
- Assume free and undampened system
  - $f(t)=0$
  - $C=0$
  - $m\ddot{u}(t) + ku(t) = 0$
- $m\ddot{u}_1 = -ku_1 + k(u_2 - u_1)$ & $m\ddot{u}_2 = -k(u_2 - u_1) + k(-u_2)$

$m_1=m_2=m$

$k_1=k_2=k_3=k$

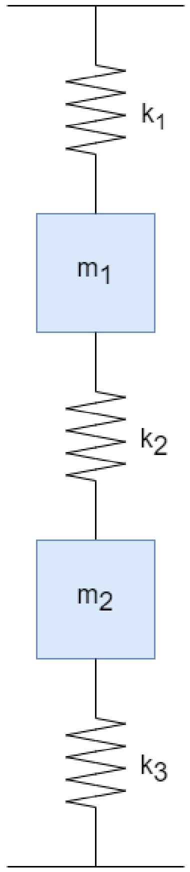


https://farside.ph.utexas.edu/teaching/315/Waveshtml/node21.html

Modal Analysis of a spring mass system

$m_1 = m_2 = m$

$k_1 = k_2 = k_3 = k$



- $m\ddot{u}_1 = -ku_1 + k(u_2 - u_1)$ & $m\ddot{u}_2 = -k(u_2 – u_1) + k(-u_2)$ (eq. 1)
- $\omega_0 = \mathrm{sqrt}(k/m)$
- $\ddot{u}_1 = -2\omega_0{}^2 u_1 + \omega_0{}^2 u_2$ & $\ddot{u}_2 = \omega_0{}^2 u_1 - 2\omega_0{}^2 u_2$ (eq. 2)
- Assume the masses oscillate in phase
- $u_1 = \hat{u}_1 \cos(\omega t-\phi)$ & $u_2 = \hat{u}_2 \cos(\omega t-\phi)$
- The normal frequencies:
- $\omega = \pm\ \omega_0$ and $\omega = \pm\ \mathrm{sqrt}(3)*\omega_0$

Modal Analysis of a spring mass system

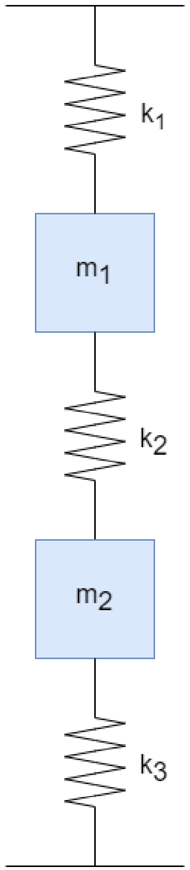
- $\omega = \pm\, \omega_0$ or $\omega = \pm$ sqrt(3)*ω_0
- The eigenvalues: $\lambda = \omega_0^2$ or $\lambda = 3*\omega_0^2$

- Find the patterns of motion:

1. $u_1 = h_1 \cos(\omega_0 t-\phi_1)$ & $u_2 = h_1 \cos(\omega_0 t-\phi_1)$

2. $u_1 = h_2 \cos(\mathrm{sqrt}(3)\omega_0 t-\phi_2)$ & $u_2 = -\, h_2 \cos(\mathrm{sqrt}(3)\omega_0 t-\phi_2)$

Arbitrary constants:
1. h_1 and ϕ_1
2. h_2 and ϕ_2



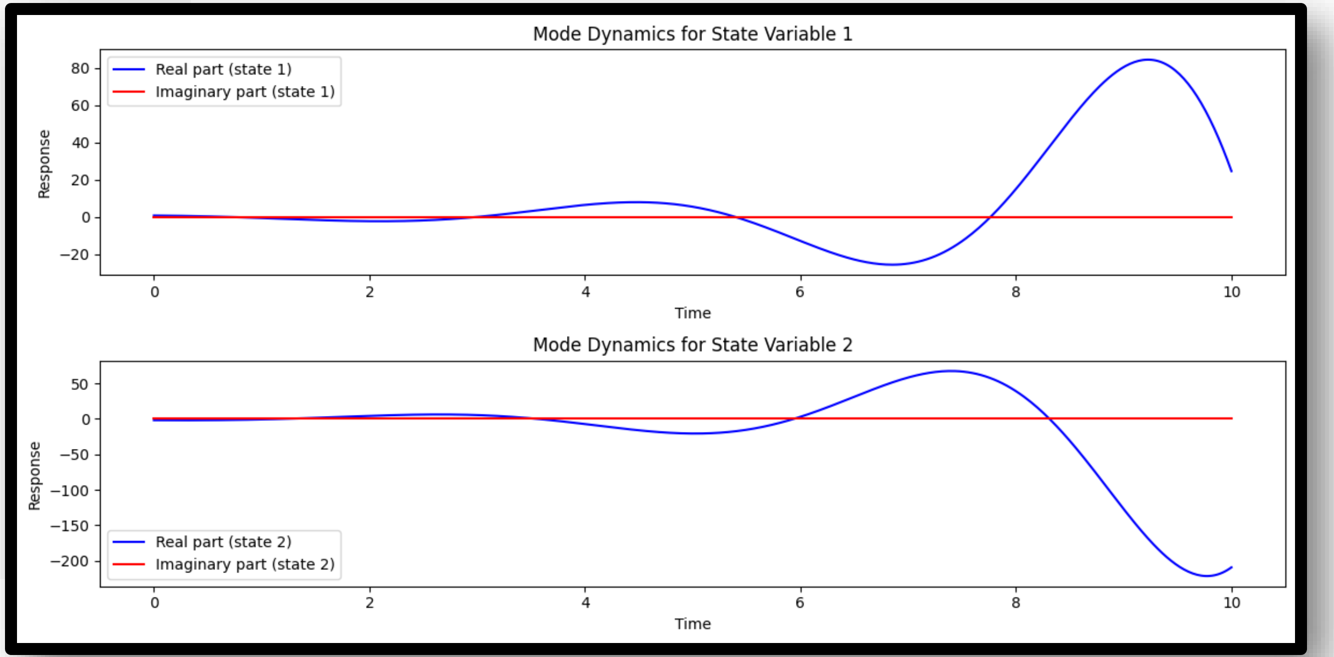
Code example in Python

```python
import numpy as np
from scipy.linalg import eig

# Define the system matrix A
A = np.array([[2, 1],
              [-4, -1]])

# Perform eigenvalue decomposition
# The scipy.linalg.eig function computes both the eigenvalues and eigenvectors of the matrix A.
# The eigenvalues provide insight into the stability and oscillatory behavior of the system,
# while the eigenvectors define the modes of the system.
eigenvalues, eigenvectors = eig(A)

# Display the eigenvalues
print("Eigenvalues:")
print(eigenvalues)

# Display the eigenvectors
print("\nEigenvectors:")
print(eigenvectors)

# Modal decomposition: each column of eigenvectors corresponds to an eigenvalue
for i in range(len(eigenvalues)):
    print(f"\nMode {i+1}:")
    print(f"Eigenvalue: {eigenvalues[i]}")
    print(f"Eigenvector: {eigenvectors[:, i]}")
```

```
Eigenvalues:
[0.5+1.32287566j 0.5-1.32287566j]

Eigenvectors:
[[ 0.3354102 +0.29580399j  0.3354102 -0.29580399j]
 [-0.89442719+0.j         -0.89442719-0.j        ]]

Mode 1:
Eigenvalue: (0.4999999999999999+1.3228756555322951j)
Eigenvector: [ 0.3354102 +0.29580399j -0.89442719+0.j        ]

Mode 2:
Eigenvalue: (0.4999999999999999-1.3228756555322951j)
Eigenvector: [ 0.3354102 -0.29580399j -0.89442719-0.j        ]
```

Code example in Python

```python
# Time array for simulation
t = np.linspace(0, 10, 500)

# Initialize array for storing system response
response = np.zeros((len(t), 2), dtype=complex)

# Define the initial condition (for example, use the first eigenvector)
initial_condition = eigenvectors[:, 0]

# Calculate the contribution of each mode over time
for i in range(len(eigenvalues)):
    # Exponential term for time evolution: e^(λt)
    exp_term = np.exp(eigenvalues[i] * t)

    # Mode contribution to the overall system response
    mode_contribution = np.outer(exp_term, eigenvectors[:, i])

    # Accumulate the response
    response += mode_contribution

# Plot the real and imaginary parts of the response for each state
plt.figure(figsize=(12, 6))
```



Relation to Time Series Forecasting

LTI systems (Linear Time-Invariant systems) and **time series** analysis are both frameworks used to model and analyze systems evolving over time:

- With LTI systems, **eigenvalues analysis** helps to identify whether the system will exhibit **stability**, **growth**, **oscillations**, **decay**.
- **Time series forecasting** methods break down the time series into components like **trend**, **seasonality**, and **noise**.

Predicting Future States in LTI systems vs. Forecasting Future Values in Time Series:

- In LTI systems, once you have the matrix A, you can predict the future states of the system by applying the matrix exponential e^{At} to the initial condition.
- In time series forecasting, models like ARIMA and GARCH use past values and residuals to forecast future values.

Summary

It's all about understanding and predicting the future behavior of a system!

- **Modal analysis** helps in understanding the inherent modes (eigenvalues and eigenvectors) of an LTI system, which determine its behavior over time.
- **Time series forecasting** models aim to capture similar dynamics by modeling past behavior to predict future behavior, often relying on trends, seasonality (oscillations), and volatility.

Sources

- https://www.researchgate.net/publication/285684454_Modal_Control_of_Linear_Time-Invariant_Discrete-Time_Systems
- https://innovationspace.ansys.com/courses/wp-content/uploads/sites/5/2020/10/Lesson2_GoverningEquationsOfModalAnalysis.pdf
- https://www.youtube.com/watch?v=0NAjoJPkp2M&t=13s
- https://www.youtube.com/watch?v=MwsWLPtiKis (MatLab code)
- https://farside.ph.utexas.edu/teaching/315/Waveshtml/node21.html (math example)