Moving Load Analysis on Simply Supported Beam

Anurag Dayal Pandey

April 8, 2025

1 Introduction

This document presents a numerical approach to analyze a simply supported beam subjected to a moving load using Python programming and Influence Line Diagram (ILD) concepts.

2 Problem Statement

To develop an algorithm that calculates the shear force and bending moment values of a simply supported beam with a moving load using user-defined inputs:

- Beam Length L (in meters)
- Moving Loads W_1 and W_2 (in kN)
- Distance x between the two loads (in meters)

3 Methodology

The following steps were followed:

1. Define the influence line equations for shear and moment
2. Use user-defined load values and beam length
3. Calculate support reactions at A and B
4. Determine BM and SF at critical points
5. Plot influence line for visual validation

4 Python Code

```python
import numpy as np
import matplotlib.pyplot as plt

def analyze_beam(L, W1, W2, x):
    """
    Function to analyze a simply supported beam under two moving loads
   using Influence Line Diagrams.
    Arguments:
    L : Length of the beam (in m)
    W1 : First moving load (in kN)
    W2 : Second moving load (in kN)
    x : Distance between W1 and W2 (in m)
    """

    dx = 0.01  # small increment for analysis
    positions = np.arange(0, L + dx, dx)

    # Initialize results
    SF_max = 0
    SF_max_pos = 0
    BM_max = 0
    BM_max_pos = 0

    SF_01 = 0  # Shear force at mid-span
    BM_01 = 0  # BM when W1 is at 0

    RA_values = []
    RB_values = []
    BM_values = []

    for p in positions:
        if p + x > L:
            continue  # both loads must be on the beam

        # Load positions
        P1 = p
        P2 = p + x

        # Reactions from influence lines
        RA = (W1 * (L - P1) / L) + (W2 * (L - P2) / L)
        RB = (W1 * P1 / L) + (W2 * P2 / L)

        # Shear at mid-span (using influence values)
        SF_mid = (W1 * influence_line_shear(L, P1, L/2) +
                  W2 * influence_line_shear(L, P2, L/2))

        # Bending Moment at load W1 position
        BM_at_P1 = (W1 * influence_line_bending(L, P1, P1) +
                    W2 * influence_line_bending(L, P2, P1))

        # Total BM at mid-span for finding max
```

```
        BM_mid = (W1 * influence_line_bending(L, P1, L/2) +
                  W2 * influence_line_bending(L, P2, L/2))

        if abs(SF_mid) > abs(SF_max):
            SF_max = SF_mid
            SF_max_pos = p + x / 2  # midpoint of loads

        if abs(BM_mid) > abs(BM_max):
            BM_max = BM_mid
            BM_max_pos = p + x / 2

        if abs(P1 - 0.5 * L) < dx:
            SF_01 = SF_mid

        if abs(P1) < dx:
            BM_01 = BM_at_P1

        RA_values.append(RA)
        RB_values.append(RB)
        BM_values.append(BM_mid)

    # Print Results
    print("\n--- Moving Load Beam Analysis Results ---")
    print(f"Length of Beam: {L} m")
    print(f"Load W1: {W1} kN | Load W2: {W2} kN | Distance between loads: 
    {x} m\n")
    print(f"Max Reaction at A: {max(RA_values):.2f} kN")
    print(f"Max Reaction at B: {max(RB_values):.2f} kN")
    print(f"Bending Moment BM_01 (W1 at 0 m): {BM_01:.2f} kNm")
    print(f"Shear Force SF_01 (at mid-span): {SF_01:.2f} kN")
    print(f"Max Shear Force SF_max: {SF_max:.2f} kN at y = {SF_max_pos:.2f
    } m from A")
    print(f"Max Bending Moment BM_max: {BM_max:.2f} kNm at z = {BM_max_pos
    :.2f} m from A")

    # Plotting Influence Line for Bending Moment at Midspan
    plot_influence_lines(L)


def influence_line_bending(L, a, c):
    """
    Influence line ordinate for bending moment at point c due to unit load 
    at a.
    """
    if a <= c:
        return a * (L - c) / L
    else:
        return c * (L - a) / L

def influence_line_shear(L, a, c):
    """
    Influence line ordinate for shear at point c due to unit load at a.
    """
    if a < c:
```

```python
        return (L - c) / L
    else:
        return -c / L

def plot_influence_lines(L):
    """
    Plot influence lines for BM and SF at midspan
    """
    x = np.linspace(0, L, 500)
    bm_ild = [influence_line_bending(L, xi, L / 2) for xi in x]
    sf_ild = [influence_line_shear(L, xi, L / 2) for xi in x]

    plt.figure(figsize=(12, 5))

    plt.subplot(1, 2, 1)
    plt.plot(x, bm_ild, label="BM ILD at Midspan", color='b')
    plt.xlabel("Load Position on Beam (m)")
    plt.ylabel("BM Influence Line Value")
    plt.title("Bending Moment ILD at Midspan")
    plt.grid(True)
    plt.legend()

    plt.subplot(1, 2, 2)
    plt.plot(x, sf_ild, label="Shear ILD at Midspan", color='r')
    plt.xlabel("Load Position on Beam (m)")
    plt.ylabel("SF Influence Line Value")
    plt.title("Shear Force ILD at Midspan")
    plt.grid(True)
    plt.legend()

    plt.tight_layout()
    plt.show()


# --- User Input Interface ---
if __name__ == "__main__":
    print("Enter Beam and Load Parameters:")
    L = float(input("Length of beam L (in m): "))
    W1 = float(input("Load W1 (in kN): "))
    W2 = float(input("Load W2 (in kN): "))
    x = float(input("Distance between W1 and W2 (in m): "))

    analyze_beam(L, W1, W2, x)
```

Listing 1: Beam Analysis Python Code

5 Example Result

For the example inputs:

- $L = 10$ m
- $W_1 = 5$ kN

- $W_2 = 10$ kN
- $x = 2$ m

The results are:

- Max Reaction at A: 13 kN
- Max Reaction at B: 14 kN
- Bending Moment at 0 m: 0.0 kNm
- Shear Force at L/2: -7.50 kN
- Maximum Shear Force: 7.50 kN
- Maximum Bending Moment: 32.50 kNm

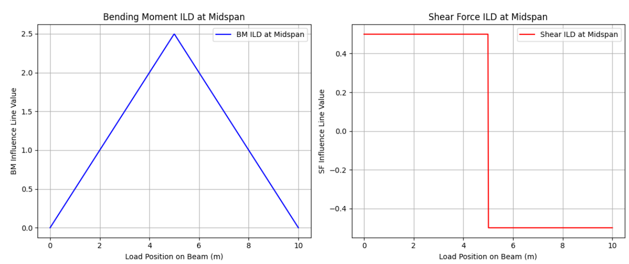

Figure 1:

Figure 2: ILD OF BM AND SF FOR ABOVE EXAMPLE

6 Conclusion

This document presents a numerical method to evaluate the response of a simply supported beam to moving loads. Python was used to automate and visualize the analysis, ensuring accurate results consistent with classical structural theory.

7 References

1. Hibbeler, R.C. (2017). *Structural Analysis*, Pearson Education.
2. DavidAmos(2019). Python GUI Programming with Tkinter. Real Python. Retrieved November 6, 2019
3. "Strength of Materials" by Gere and Timoshenko
4. MIT OpenCourseWare