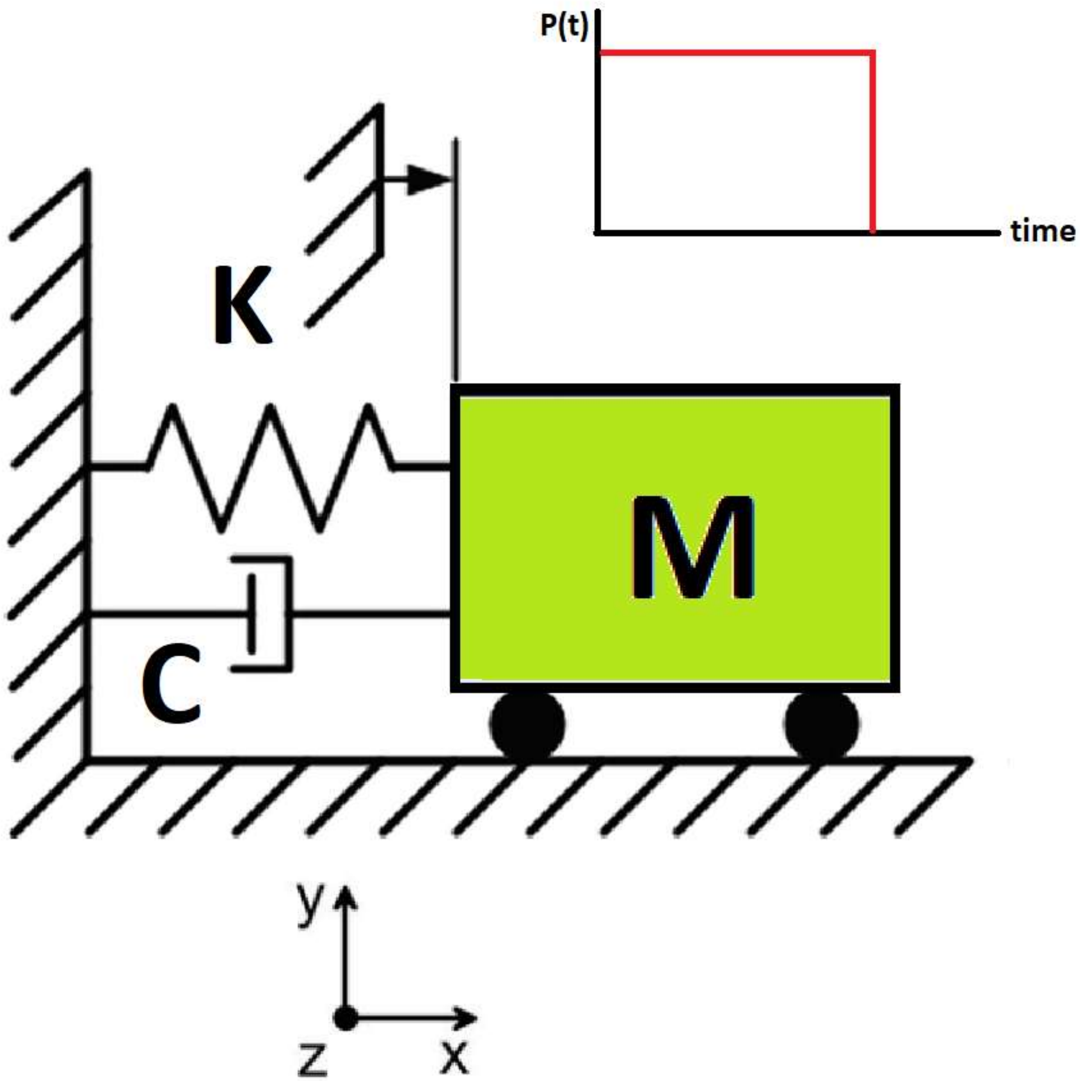
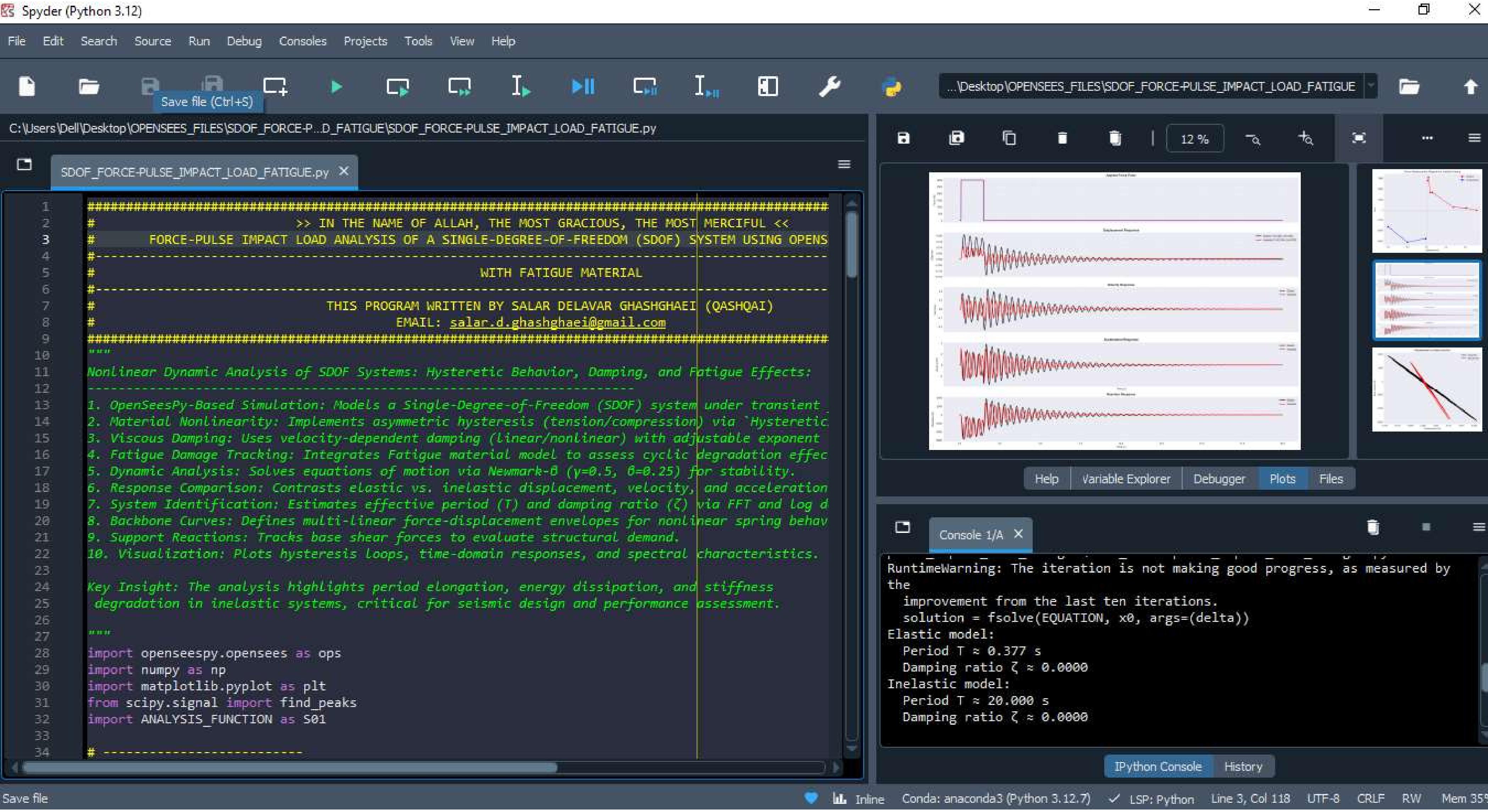
>> IN THE NAME OF ALLAH, THE MOST GRACIOUS, THE MOST MERCIFUL <<

FORCE-PULSE IMPACT LOAD ANALYSIS OF A SINGLE-DEGREE-OF-FREEDOM (SDOF) SYSTEM USING OPENSEES

WRITTEN BY SALAR DELAVAR GHASHGHAEI (QASHQAI)


P(t)
time
K
M
C
y
z
x



Spyder (Python 3.12)
File Edit Search Source Run Debug Consoles Projects Tools View Help
Save file (Ctrl+S)
...\Desktop\OPENSEES_FILES\SDOF_FORCE-PULSE_IMPACT_LOAD_FATIGUE
C:\Users\Dell\Desktop\OPENSEES_FILES\SDOF_FORCE-P...D_FATIGUE\SDOF_FORCE-PULSE_IMPACT_LOAD_FATIGUE.py
SDOF_FORCE-PULSE_IMPACT_LOAD_FATIGUE.py
>> IN THE NAME OF ALLAH, THE MOST GRACIOUS, THE MOST MERCIFUL <<
FORCE-PULSE IMPACT LOAD ANALYSIS OF A SINGLE-DEGREE-OF-FREEDOM (SDOF) SYSTEM USING OPENS
WITH FATIGUE MATERIAL
THIS PROGRAM WRITTEN BY SALAR DELAVAR GHASHGHAEI (QASHQAI)
EMAIL: salar.d.ghashghaei@gmail.com
Nonlinear Dynamic Analysis of SDOF Systems: Hysteretic Behavior, Damping, and Fatigue Effects:
1. OpenSeesPy-Based Simulation: Models a Single-Degree-of-Freedom (SDOF) system under transient
2. Material Nonlinearity: Implements asymmetric hysteresis (tension/compression) via `Hysteretic
3. Viscous Damping: Uses velocity-dependent damping (linear/nonlinear) with adjustable exponent
4. Fatigue Damage Tracking: Integrates Fatigue material model to assess cyclic degradation effec
5. Dynamic Analysis: Solves equations of motion via Newmark-β (γ=0.5, β=0.25) for stability.
6. Response Comparison: Contrasts elastic vs. inelastic displacement, velocity, and acceleration
7. System Identification: Estimates effective period (T) and damping ratio (ζ) via FFT and log d
8. Backbone Curves: Defines multi-linear force-displacement envelopes for nonlinear spring behav
9. Support Reactions: Tracks base shear forces to evaluate structural demand.
10. Visualization: Plots hysteresis loops, time-domain responses, and spectral characteristics.
Key Insight: The analysis highlights period elongation, energy dissipation, and stiffness
degradation in inelastic systems, critical for seismic design and performance assessment.
import openseespy.opensees as ops
import numpy as np
import matplotlib.pyplot as plt
from scipy.signal import find_peaks
import ANALYSIS_FUNCTION as S01
# --------------------------
12 %
Help Variable Explorer Debugger Plots Files
Console 1/A
RuntimeWarning: The iteration is not making good progress, as measured by the
improvement from the last ten iterations.
solution = fsolve(EQUATION, x0, args=(delta))
Elastic model:
Period T ≈ 0.377 s
Damping ratio ζ ≈ 0.0000
Inelastic model:
Period T ≈ 20.000 s
Damping ratio ζ ≈ 0.0000
IPython Console History
Save file
Inline Conda: anaconda3 (Python 3.12.7) LSP: Python Line 3, Col 118 UTF-8 CRLF RW Mem 35%

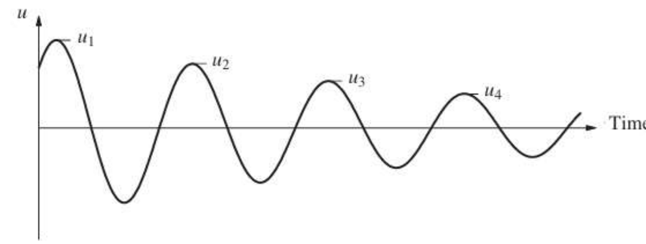
VISCOUSLY DAMPED FREE VIBRATION

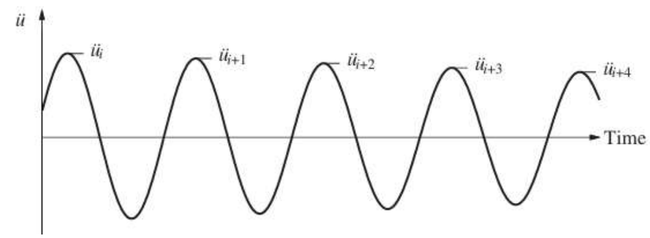
$$m\ddot{u} + c\dot{u} + ku = 0$$

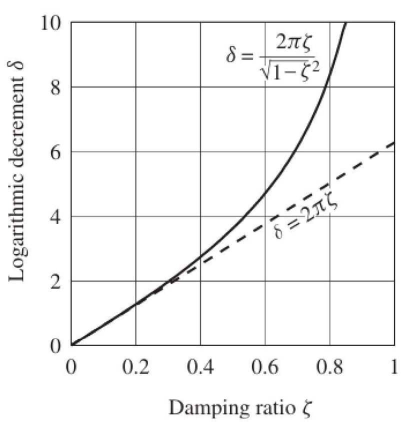
$$\ddot{u} + 2\zeta\omega_n\dot{u} + \omega_n^2 u = 0$$

$$\omega_n = \sqrt{k/m} \quad \zeta = \frac{c}{2m\omega_n} = \frac{c}{c_{cr}} \quad \omega_D = \omega_n\sqrt{1-\zeta^2}$$

$$u(t) = e^{-\zeta\omega_n t}\left[u(0)\cos\omega_D t + \frac{\dot{u}(0) + \zeta\omega_n u(0)}{\omega_D}\sin\omega_D t\right]$$





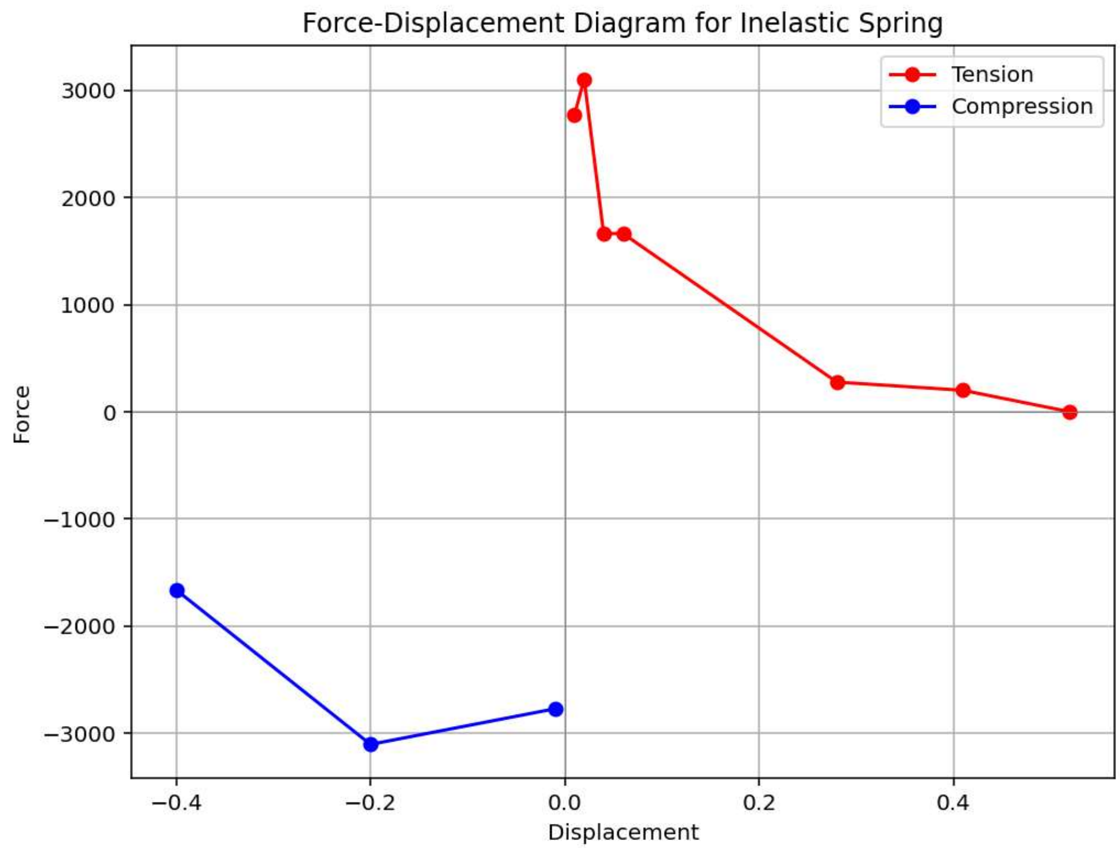


Decay of Motion

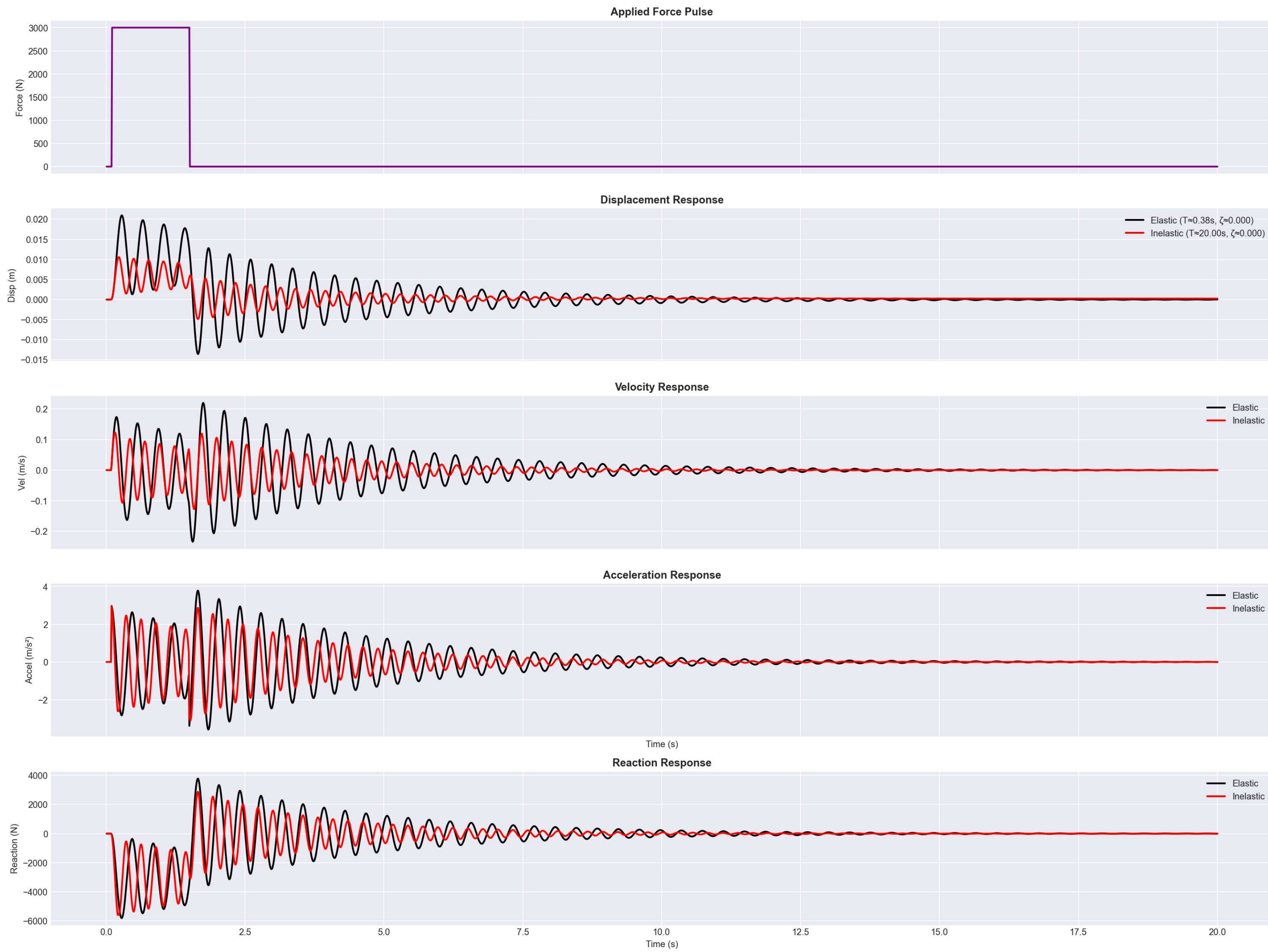
$$\delta = \ln\frac{u_i}{u_{i+1}} = 2\pi\zeta$$ **(APPROXIMATE RELATION)**

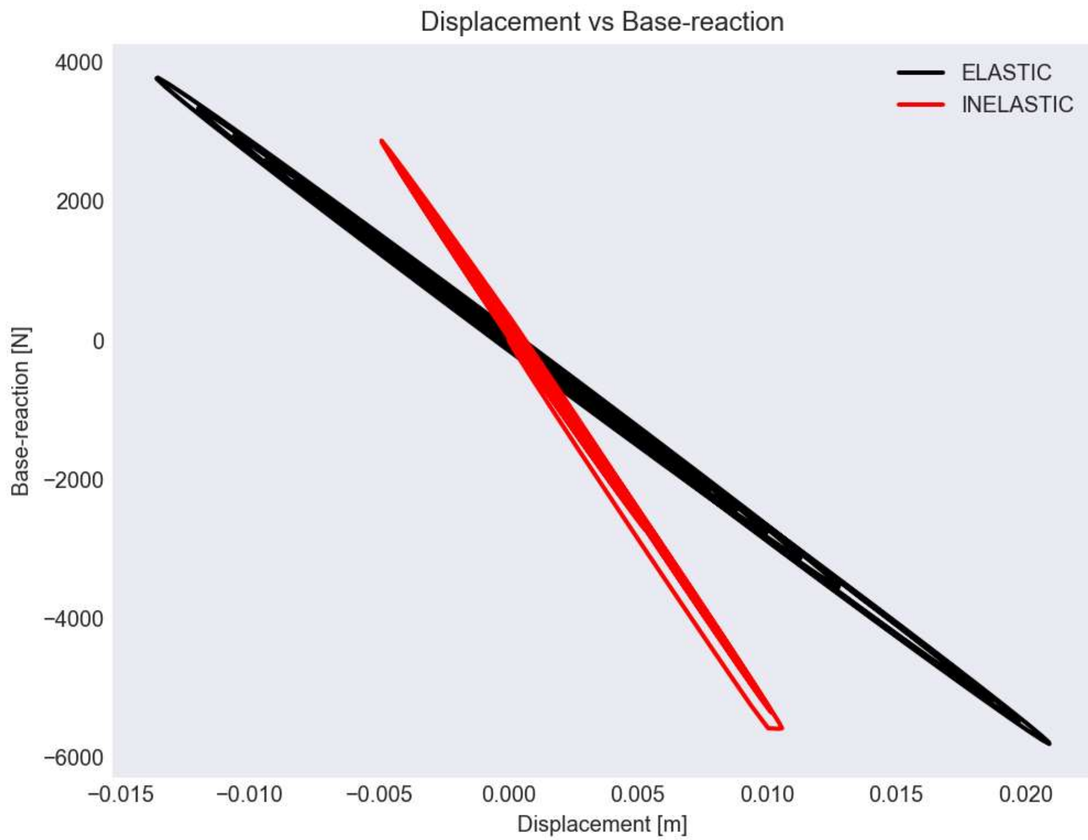
$$\delta = \ln\frac{u_i}{u_{i+1}} = \frac{2\pi\zeta}{\sqrt{1-\zeta^2}}$$ **(EXACT RELATION)**

EXACT AND APPROXIMATE RELATIONS BETWEEN LOGARITHMIC DECREMENT AND DAMPING RATIO



Force-Displacement Diagram for Inelastic Spring
Tension
Compression
3000
2000
1000
0
−1000
−2000
−3000
Force
−0.4
−0.2
0.0
0.2
0.4
Displacement


Applied Force Pulse
Force (N)
Displacement Response
Disp (m)
Elastic (T≈0.38s, ζ≈0.000)
Inelastic (T≈20.00s, ζ≈0.000)
Velocity Response
Vel (m/s)
Elastic
Inelastic
Acceleration Response
Accel (m/s²)
Time (s)
Reaction Response
Reaction (N)
Time (s)


Displacement vs Base-reaction
ELASTIC
INELASTIC
4000
2000
0
−2000
−4000
−6000
Base-reaction [N]
−0.015
−0.010
−0.005
0.000
0.005
0.010
0.015
0.020
Displacement [m]