

PINN for Earthquake Response, Material Behaviour, and Robotic Inspection

Physics-Informed Neural Networks Applied to Structural Dynamics, Construction Materials, and Robotics

Project brief

Objective: Build Physics-Informed Neural Networks that solve governing equations across three engineering domains — structural dynamics, construction materials, and robotics. Validate each against analytical or numerical benchmarks, then reflect on the strengths and limitations of each approach.

Structural Parameters

- Natural period $T_n = 0.5$ s
- Damping ratio $\zeta = 0.05$ (5%)
- Normalised mass $m = 1$ (mass-normalised formulation)
- Derived: $\omega_n = 2\pi/T_n = 4\pi$ rad/s

Ground Motion

- Record: El Centro 1940 NS component
- Duration: first 15 s (strong motion)
- Sampling: $\Delta t = 0.02$ s (750 points)
- Peak ground acceleration: $0.319g$

Open-ended project and use of AI assistants: This project is designed as an open-ended exercise. The lecture notes may not fully cover all the questions; students are encouraged to explore beyond the course material to complete each step. You are also encouraged to use AI coding assistants (e.g. ChatGPT, Claude, GitHub Copilot and others) to help design and debug your Python code. However, you must understand every line you submit. In your PDF report, briefly state which AI tool(s) you used and how they helped.

Submission: Python files (.py) or Excel files and a PDF report (containing all figures) submitted via email to wenhui.duan@monash.edu. Deadline: 5:00 pm Beijing time (UTC+8), Monday 6 April 2026.

Project steps

Step 1: Forward PINN — SDOF earthquake response (1.5 marks)

Purpose: Embed the earthquake equation of motion into a PINN loss function and train the network to predict displacement without any training data.

Governing equation (normalised form):

$$\ddot{u} + 2\zeta\omega_n \dot{u} + \omega_n^2 u = -\ddot{u}_g(t)$$

where $u(t)$ is relative displacement, $\zeta = 0.05$, $\omega_n = 4\pi$ rad/s, and $\ddot{u}_g(t)$ is recorded ground acceleration. Initial conditions: $u(0) = 0$, $\dot{u}(0) = 0$.

PINN formulation:

- **Input:** time $t \in [0, 15]$ s
- **Output:** predicted displacement $\hat{u}(t)$
- **Derivatives:** \hat{u} , $\ddot{\hat{u}}$ via automatic differentiation (`torch.autograd`)
- **Ground acceleration:** interpolate the discrete record using `scipy.interpolate.interp1d` to evaluate at arbitrary collocation points
- **ODE residual** at each collocation point t_i :

$$R(t_i) = \ddot{\hat{u}}(t_i) + 2\zeta\omega_n \dot{\hat{u}}(t_i) + \omega_n^2 \hat{u}(t_i) + \ddot{u}_g(t_i)$$

- **Loss function:**

$$\mathcal{L} = \lambda_{\text{PDE}} \frac{1}{N_c} \sum_{i=1}^{N_c} R(t_i)^2 + \lambda_{\text{IC}} \left[\hat{u}(0)^2 + \dot{\hat{u}}(0)^2 \right]$$

Implementation tip: In your code, normalise the input time to $[0, 1]$ before feeding it to the network (i.e. $\tilde{t} = t/T_{\text{max}}$). This helps training converge faster. Remember to scale the derivatives back to physical units when computing the ODE residual. The formulation above is written in physical time for clarity.

Suggested hyperparameters:

Parameter	Value	Note
Hidden layers	4	Same depth as dl-basics crane PINN
Neurons per layer	64	Sufficient for single-period response
Activation	tanh	Smooth; required for second derivatives
Collocation points	2000	Uniformly spaced over $[0, 15]$ s
Optimiser (phase 1)	Adam, $\eta = 10^{-3}$	20 000 epochs
Optimiser (phase 2)	L-BFGS	5 000 iterations (fine-tuning)
λ_{PDE}	1.0	
λ_{IC}	100.0	High weight to anchor initial conditions

Deliverables:

- Training loss curve (log scale) vs epoch
- Displacement $\hat{u}(t)$ vs time plot
- Print peak displacement to console

Step 2: Newmark benchmark and validation (1.5 marks)

Purpose: Validate the PINN against the established Newmark numerical integration method taught in CIV5885.

Implement Newmark's **constant-average-acceleration** method ($\beta = 1/4$, $\gamma = 1/2$, $\Delta t = 0.02$ s) for the same SDOF system and ground motion.

Deliverables:

- Overlay plot: PINN displacement vs Newmark displacement vs time
- Error plot: $|\hat{u}_{\text{PINN}}(t) - u_{\text{Newmark}}(t)|$ vs time
- Peak displacement comparison table
- **Sensitivity study:** Train separate PINNs with 200, 500, 1000, and 2000 collocation points. For each, report the peak displacement error relative to Newmark. Present results as a bar chart.

Hint: Newmark's constant-average-acceleration method ($\beta = 1/4$) is **unconditionally stable** for any Δt . The PINN has no explicit time step — its resolution depends on the

density of collocation points.

Step 3: Material PINN — viscoelastic creep response (2.0 marks)

Purpose: Apply the PINN methodology from Step 1 to a construction materials ODE. Solve the Kelvin–Voigt viscoelastic creep equation and compare with the analytical solution.

Governing equation:

The Kelvin–Voigt model for transient creep under constant stress σ_0 :

$$\eta \dot{\varepsilon} + E \varepsilon = \sigma_0 \quad (1)$$

Analytical solution:

$$\varepsilon(t) = \frac{\sigma_0}{E} \left[1 - \exp\left(-\frac{E}{\eta} t\right) \right] \quad (2)$$

Material parameters:

Material	E (MPa)	η (MPa·s)	$\tau = \eta/E$ (s)	σ_0 (MPa)	$\varepsilon(\infty)$
Bitumen (40°C)	50	5 000	100	0.5	0.01
Polymer (room temp.)	500	50 000	100	5.0	0.01
Softwood timber	8 000	800 000	100	2.0	0.00025

PINN formulation:

- **Input:** time $t \in [0, 500]$ s
- **Output:** predicted strain $\hat{\varepsilon}(t)$
- **ODE residual** at each collocation point t_i :

$$R(t_i) = \eta \dot{\hat{\varepsilon}}(t_i) + E \hat{\varepsilon}(t_i) - \sigma_0$$

- **Initial condition:** $\hat{\varepsilon}(0) = 0$
- **Loss function:**

$$\mathcal{L} = \lambda_{\text{ODE}} \frac{1}{N_c} \sum_{i=1}^{N_c} R(t_i)^2 + \lambda_{\text{IC}} \hat{\varepsilon}(0)^2$$

Network: Same as Step 1 (4 hidden layers \times 64 neurons, tanh activation).

Optimiser: Adam ($\eta = 10^{-3}$, 10 000 epochs) + L-BFGS (3 000 iterations).

Collocation points: $N_c = 500$ uniformly in $[0, 500]$ s.

Submission requirements:

- Python file (.py) with all code and comments
- Figure 1: PINN vs analytical solution overlay (3 subplots, one per material)
- Figure 2: Training loss curve (total, data, physics components)
- Table 1: Peak absolute error and relative error for each material
- Discussion (100–200 words): How does the time constant τ affect training difficulty?

Reference solutions

- Bitumen: $\varepsilon(0) = 0$, $\varepsilon(\infty) = 0.01$, $\varepsilon(100) = 0.01(1 - e^{-1}) = 0.00632$
- Polymer: $\varepsilon(\infty) = 0.01$, $\varepsilon(100) = 0.00632$
- Timber: $\varepsilon(\infty) = 0.00025$, $\varepsilon(100) = 0.000158$
- Expected PINN error: $< 1\%$ relative after convergence
- Expected training: $\sim 5\,000$ epochs (first-order ODE easier than Step 1's second-order)

Step 4: Robot kinematic analysis of inspection path (2.0 marks)

Purpose: Use a 2R planar arm as a simplified model for a structural inspection robot. Compute the joint trajectories and torques needed to trace a straight-line inspection path along an earthquake-damaged beam.

Setup:

2R arm parameters:

- $L_1 = 1.0$ m, $L_2 = 0.5$ m
- $m_1 = 2.0$ kg, $m_2 = 1.0$ kg
- $L_{c1} = 0.5$ m, $L_{c2} = 0.25$ m
- $I_1 = 0.5$ kg·m², $I_2 = 0.0625$ kg·m²

Inspection path: straight line from $A = (1.0, 0.5)$ to $B = (0.5, 1.2)$, representing a beam-column joint.

Complete the following sub-tasks (a–g). The equations below are provided for reference; you may use Python or Excel.

(a) Verify workspace reachability:

$$r_A = \sqrt{1.0^2 + 0.5^2} = 1.118 \text{ m}, \quad r_B = \sqrt{0.5^2 + 1.2^2} = 1.300 \text{ m}$$

Both within $[0.5, 1.5]$. ✓

(b) Compute inverse kinematics (IK) at both endpoints (elbow-up):

Forward kinematics:

$$x = L_1 \cos \theta_1 + L_2 \cos(\theta_1 + \theta_2), \quad y = L_1 \sin \theta_1 + L_2 \sin(\theta_1 + \theta_2)$$

Inverse kinematics:

$$\cos \theta_2 = \frac{x^2 + y^2 - L_1^2 - L_2^2}{2 L_1 L_2}, \quad \theta_1 = \text{atan2}(y, x) - \text{atan2}(L_2 \sin \theta_2, L_1 + L_2 \cos \theta_2)$$

(c) Generate straight-line Cartesian path with $N = 50$ points:

$$x(s) = 1.0 - 0.5 s, \quad y(s) = 0.5 + 0.7 s, \quad s \in [0, 1]$$

(d) Apply IK at each point $\rightarrow \theta_1(s), \theta_2(s)$.

(e) Cubic time scaling:

$$s(t) = \frac{3 t^2}{T^2} - \frac{2 t^3}{T^3}, \quad T = 5 \text{ s}$$

(f) Joint velocities:

$$\dot{\theta} = \frac{d\theta}{ds} \dot{s}(t)$$

(g) Inverse dynamics torques:

$$\boldsymbol{\tau} = \mathbf{M}(\boldsymbol{\theta}) \ddot{\boldsymbol{\theta}} + \mathbf{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \dot{\boldsymbol{\theta}} + \mathbf{g}(\boldsymbol{\theta})$$

Mass matrix elements:

$$M_{11} = m_1 L_{c1}^2 + I_1 + m_2 (L_1^2 + L_{c2}^2 + 2 L_1 L_{c2} \cos \theta_2) + I_2$$

$$M_{12} = m_2 (L_{c2}^2 + L_1 L_{c2} \cos \theta_2) + I_2$$

$$M_{22} = m_2 L_{c2}^2 + I_2$$

Submission requirements:

- Python file (.py) or Excel spreadsheet (.xlsx)
- Figure 1: Cartesian path (x vs y) with arm configurations shown at 5 evenly spaced positions

- Figure 2: $\theta_1(t)$ and $\theta_2(t)$ vs time
- Figure 3: $\tau_1(t)$ and $\tau_2(t)$ vs time
- Table 1: IK solutions at endpoints A and B
- Report: $\min |\det(\mathbf{J})|$ along path; assess singularity proximity

Reference solutions

- IK at A : $\cos \theta_2 = (1.25 - 1.25)/1.0 = 0 \rightarrow \theta_2 = 90^\circ$; $\theta_1 = \text{atan2}(0.5, 1.0) - \text{atan2}(0.5, 1.0) = 0^\circ$
Verify: $x = \cos 0^\circ + 0.5 \cos 90^\circ = 1.0$, $y = \sin 0^\circ + 0.5 \sin 90^\circ = 0.5 \checkmark$
- IK at B : $\cos \theta_2 = (1.69 - 1.25)/1.0 = 0.44 \rightarrow \theta_2 = 63.9^\circ$
- Peak torques within $[20, 10]$ Nm limits
- $\min |\det(\mathbf{J})| > 0.2$ (no near-singularity)

Step 5: Two-input PINN for response spectrum (1.5 marks)

Purpose: Extend the PINN to learn the entire family of SDOF responses across a range of natural periods, then extract the earthquake response spectrum.

Key idea: A standard response spectrum requires a separate numerical integration for each natural period T_n (typically 50–100 runs). A two-input PINN learns all these solutions *simultaneously* by treating T_n as a second input.

Extended PINN formulation:

- **Input:** (t, T_n) where $t \in [0, 15]$ s and $T_n \in [0.1, 3.0]$ s
- **Output:** $\hat{u}(t, T_n)$
- **Period range:** $T_n \in [0.1, 3.0]$ s (50 values)
- **ODE residual** at each collocation point $(t_i, T_{n,j})$:

$$R(t_i, T_{n,j}) = \ddot{u} + 2\zeta \frac{2\pi}{T_{n,j}} \dot{u} + \left(\frac{2\pi}{T_{n,j}} \right)^2 \hat{u} + \ddot{u}_g(t_i)$$

where $\zeta = 0.05$ for all oscillators (standard for spectrum construction).

- **Collocation:** 150 time points \times 30 period points = 4500 total
- **Initial conditions:** For every $T_{n,j}$: $\hat{u}(0, T_{n,j}) = 0$ and $\dot{\hat{u}}(0, T_{n,j}) = 0$

Larger network (5 layers \times 128 neurons) is needed because the network must represent a family of solutions parameterised by T_n .

Spectrum extraction after training:

$$D(T_n) = \max_t |\hat{u}(t, T_n)|, \quad V(T_n) = \frac{2\pi}{T_n} D(T_n), \quad A(T_n) = \left(\frac{2\pi}{T_n}\right)^2 D(T_n)$$

Benchmark: Compute the same spectrum using Newmark integration for each of the 50 periods.

Deliverables:

- Three spectrum plots: $D(T_n)$, $V(T_n)$, $A(T_n)$ — PINN vs Newmark on each
- Percentage error in peak pseudo-acceleration

Step 6: Integration and design reflection (1.5 marks)

Purpose: Bring together the results from all five previous steps. Compare the PINN-generated response spectrum against the Australian Standard design spectrum, assess material damage and robot feasibility, and reflect on the different computational approaches.

(a) AS 1170.4 design spectrum comparison

AS 1170.4 design pseudo-acceleration for Site Class D_e , $Z = 0.10$, $k_p = 1.0$:

$$A_{\text{design}}(T) = k_p Z C_h(T) \times g$$

where $C_h(T)$ is the spectral shape factor. Use the simplified Site Class D_e values below (from AS 1170.4 Table 6.4):

Period range	$C_h(T)$	Note
$T \leq 0.0$ s	1.10	Zero-period acceleration
$0 < T \leq 0.1$ s	Linear from 1.10 to 3.68	Rising branch
$0.1 < T \leq 1.5$ s	$3.68 \times (0.1/T)^{0.75}$	Descending branch
$T > 1.5$ s	$3.68 \times (0.1/1.5)^{0.75} \times (1.5/T)$	Displacement-controlled

Example: At $T = 0.5$ s, $C_h = 3.68 \times (0.1/0.5)^{0.75} = 1.04$. Design acceleration = $1.0 \times 0.10 \times 1.04 \times 9.81 = 1.02 \text{ m/s}^2$ (0.104g).

- **Combined plot:** Pseudo-acceleration spectrum from Step 5 (PINN and Newmark) overlaid with the AS 1170.4 design spectrum

- **Annotate three spectral regions** on the plot:
 - Short period ($T < 0.3$ s): acceleration-sensitive
 - Intermediate ($0.3 < T < 1.5$ s): velocity-sensitive
 - Long period ($T > 1.5$ s): displacement-sensitive
- **Mark $T = 0.82$ s** on the plot (the Adelaide office building from the Seismic Design Project)

Connection to Task 1: At $T = 0.82$ s, read off the pseudo-acceleration A from your PINN spectrum. Compute the equivalent base shear $V = A \times (W_t/g) = A \times (19,000/9.81)$. Compare this record-specific value with the AS 1170.4 design base shear of 874 kN from the Seismic Design Project.

(b) Material damage assessment

At the peak displacement from Step 1, compute the peak strain in a concrete column with height $h = 3.5$ m. The drift ratio is u_{\max}/h . Compare this drift ratio to the creep strains computed in Step 3. Is the earthquake-induced strain larger or smaller than long-term creep?

(c) Robot inspection feasibility

From Step 4, report total time T , peak joint velocity, and peak joint torque. Is the inspection path feasible for a robot with torque limits $\tau_{\max} = [20, 10]$ Nm?

(d) Written reflection (300–500 words)

Compare analytical (Newmark), numerical (Euler integration for robot dynamics), and machine learning (PINN) approaches across all steps. Which is most accurate? Which is most flexible? Where does each approach struggle?

Submission requirements:

- Combined PDF report
- All figures from Steps 1–5 collected and cross-referenced
- Reflection addresses: accuracy, flexibility, computational cost

Reference answers for reflection

- Newmark is most accurate for structural dynamics
- PINN is most flexible (same framework solves different ODEs)
- Analytical IK is fastest where closed-form exists
- PINNs struggle with stiff ODEs, high-frequency content, λ tuning

Summary of project steps

Step	Topic	Domain	Marks
1	Forward PINN (SDOF earthquake)	Dynamics and materials	1.5
2	Newmark benchmark & sensitivity	Dynamics and materials	1.5
3	Material PINN (Kelvin–Voigt creep)	Construction materials	2.0
4	Robot inspection path (IK + dynamics)	Robotics	2.0
5	Two-input PINN (response spectrum)	Structural dynamics	1.5
6	Integration & design reflection	All three domains	1.5
Total			10.0

Marking criteria (10 marks)

Criterion	Marks	Key indicators
Step 1: PINN formulation	1.5	Correct ODE residual, IC loss, ground motion interpolation, converged training
Step 2: Newmark benchmark	1.5	Correct Newmark implementation, overlay plot, sensitivity study
Step 3: Material PINN	2.0	Kelvin–Voigt ODE residual, three-material comparison, error analysis
Step 4: Robot inspection	2.0	Correct IK, Cartesian path, torque computation, singularity assessment
Step 5: Response spectrum	1.5	Two-input PINN, spectrum extraction, PINN vs Newmark comparison
Step 6: Integration & reflection	1.5	Design spectrum overlay, material damage, robot feasibility, written reflection
Total	10.0	

Overall submission format

- **Deadline:** 5:00 pm Beijing time (UTC+8), Monday 6 April 2026
 - **Submission:** Email to `wenhui.duan@monash.edu`
 - **Format:** Python source files (`.py`) for all steps, plus one PDF report containing all 11 figures/images with captions and brief discussion. No `.ipynb` files.
 - **Group size:** Individual or group (any size). Students unfamiliar with Python may team up with experienced members, or concentrate on hand-calculation steps (e.g. Step 4 via Excel). Each group submits one set of files. Marking is individual-based: group marks are averaged by the number of group members.
 - **Student ID:** Every submission (email and PDF report) must include the student ID(s) of all members. Marks will be released based on student ID only.
 - **Total:** 10 marks, minimum 4/10 to pass this component
-

Student review questions

Q1. Mass normalisation.

The normalised equation $\ddot{u} + 2\zeta\omega_n\dot{u} + \omega_n^2u = -\ddot{u}_g(t)$ contains no mass term. Explain why two SDOF systems with different masses but the same ω_n and ζ produce identical relative displacement $u(t)$ under the same ground motion. How does this simplify the PINN formulation?

Hint: Divide the full equation $m\ddot{u} + c\dot{u} + ku = -m\ddot{u}_g$ by m . All mass terms cancel, leaving only $\omega_n = \sqrt{k/m}$ and $\zeta = c/(2m\omega_n)$. The PINN therefore only needs ω_n and ζ as parameters, not m , c , k separately. Refer to Chopra §6.1.1.

Q2. PINN vs Newmark: what is the “time step”?

Newmark’s constant-average-acceleration method ($\beta = 1/4$) is unconditionally stable for any Δt . The PINN has no explicit time step. Discuss: what controls the temporal resolution of the PINN solution? What happens to accuracy when you reduce collocation points from 2000 to 200?

Hint: The collocation point density plays a role analogous to the time step. Fewer points mean the ODE residual is enforced at fewer locations, allowing the network to “cheat” between collocation points. Unlike Newmark, the PINN error is not controlled by a stability criterion but by the richness of the loss landscape.

Q3. Comparing PINN training across ODE orders.

You solved two different ODEs with PINNs (earthquake: second-order, creep: first-order). Compare the training difficulty. Which converged faster? Why? Discuss how the ODE order, number of initial conditions, and forcing functions affect PINN training.

Hint: The first-order creep ODE requires only one initial condition and one derivative, whereas the second-order earthquake ODE requires two initial conditions and second-order derivatives via automatic differentiation. Higher-order derivatives amplify gradient noise during training. The creep ODE also has a smooth monotonic solution, making it easier for the network to approximate.

Q4. Coupled structural–robotic PINN.

In Step 4, the robot dynamics are computed independently of the structural response. In reality, a heavy inspection robot would affect the structure. How would you formulate a coupled PINN that simultaneously solves the structural response and the robot’s trajectory optimisation?

Hint: A coupled formulation would require a multi-output PINN with shared inputs (time) and separate output heads for structural displacement and joint angles. The loss function would include both the structural ODE residual and the robot dynamics equations, plus a coupling term representing the reaction forces between the robot and the structure. This is analogous to multi-physics PINNs used in fluid–structure interaction problems.

Q5. From response spectrum to design spectrum.

Your Step 5 PINN computes a response spectrum for *one* earthquake record. A design spectrum (AS 1170.4 Table 6.4) is a smooth envelope from statistical analysis of *many* records. Propose how you would extend the PINN approach to generate a design spectrum. What additional network inputs would be needed? What training data would be required?

Hint: One approach: add a third input representing the earthquake record identity (e.g., an index or a set of ground-motion intensity measures such as PGA, PGV, and duration). Train on many records simultaneously. The network learns a mapping

from $(t, T_n, \text{record features}) \rightarrow u$. The design spectrum would then be the statistical envelope (e.g., mean + 1 standard deviation) over all records. Refer to CIV5885 Item 28.