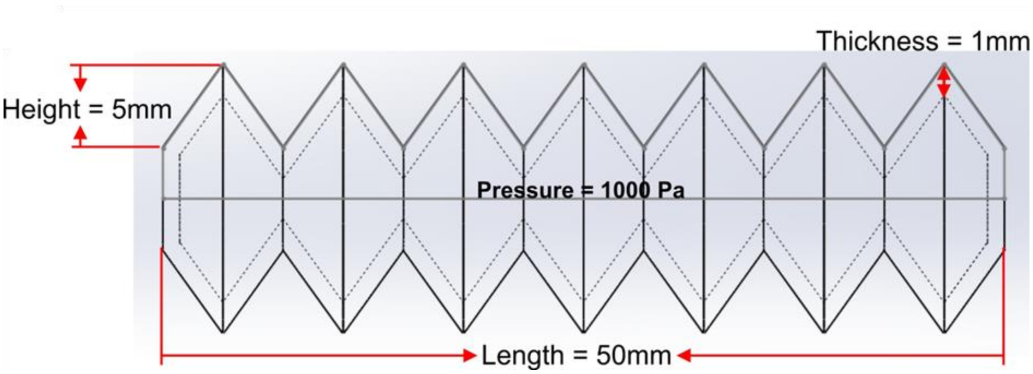
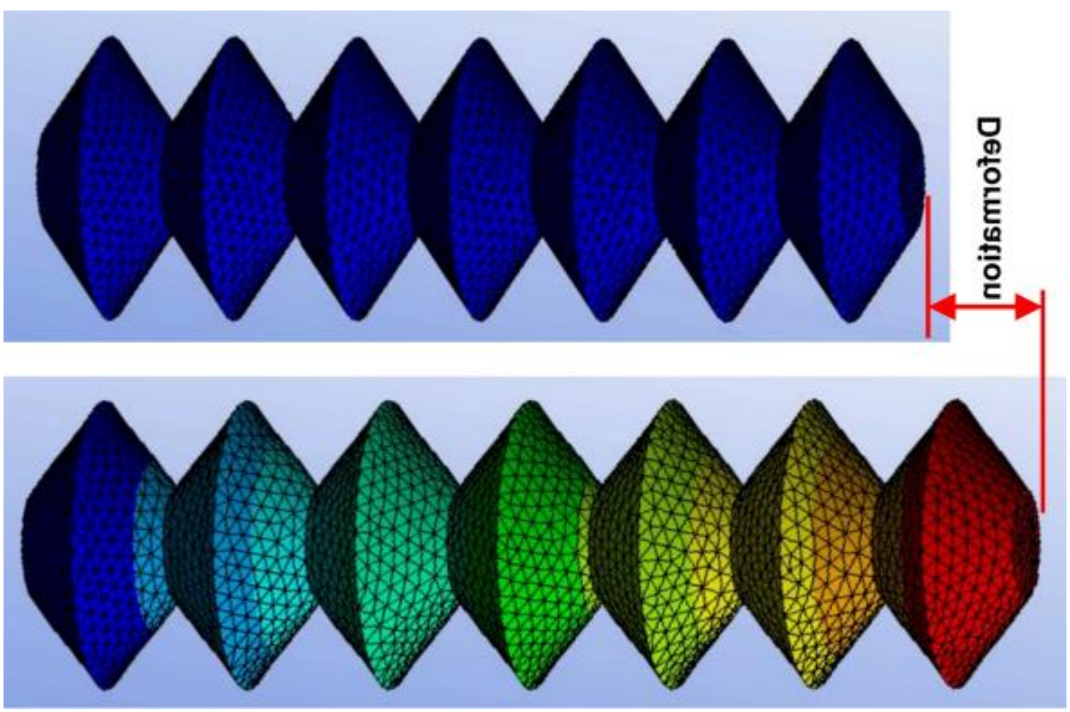
# Assignment: Data-Driven Surrogate Model for Deformation of a Soft Bellow Actuator

A soft bellow actuator has been modelled using a hyperelastic material model and simulated in ANSYS. The actuator design is described by four geometric / operating parameters:

## Pressure, Height, Length, Thickness



For each combination of these parameters the ANSYS FEA returned a single scalar output: Deformation (mm). To explore design space and find parameter settings that produce a desired deformation, >1000 simulations were run and compiled into an Excel/CSV file. However, the FEA is time-consuming and there remain parameter combinations (e.g., values between grid points) that have not been simulated. Your assignment is to train and validate a machine-learning regression model that can predict deformation for arbitrary combinations of the input parameters, effectively a fast surrogate model that obviates the need to re-run FEA for each new design point.

**Simulation parameter grid**

| Parameter | Min | Max | Interval |
|---|---|---|---|
| Pressure (units) | 1000 | 5000 | 500 |
| Height (mm) | 5.0 | 8.0 | 0.5 |
| Length (mm) | 50 | 100 | 10 |
| Thickness (mm) | 1.0 | 1.2 | 0.1 |

**This full factorial grid yields 9 × 7 × 6 × 3 = 1,134 unique combinations. The provided dataset contains 1,134 unique combos.**

**Example of "missing" combination(s)**

Because parameters were sampled at discrete intervals, intermediate values are not present in the file. Example missing values you may need to predict:

- Pressure = **1250** (not in 1000,1500,...)
- Height = **5.25**
- Length = **55** (if grid only had 50,60, etc.)
- Thickness = **1.05**

Your surrogate model must give deformation estimates for such in-between values without re-running FEA.

## Assignment objective

Train a regression model (or an ensemble of models) that predicts Deformation from the four inputs, with good generalization to un-simulated input combinations. Validate, compare models, select the best, and produce a technical report documenting methods, results, and recommendations.

## Recommended ML approach (rationale & suggestions)

Given the dataset size (~1,100 samples), discrete but numeric inputs, and moderate complexity of the mapping:

1. **Primary approaches to try (must implement at least two):**
    - **Tree-based ensemble methods** e.g., Random Forest, Gradient Boosting (fitrensemble in MATLAB / TreeBagger / fitrboost). These handle nonlinearities/interactions well and require little preprocessing.
    - **Gaussian Process Regression (GPR)**, provides predictive mean and uncertainty; good for small datasets and for quantifying prediction confidence.
    - **Feedforward Neural Network (MLP)**, small net acceptable; risk of overfitting but useful as a comparison.
    - **(Optional)** Support Vector Regression (SVR) or k-NN regression.
2. **Why these?**
    - Ensemble trees excel on tabular data, need little scaling, and produce feature importance.
    - GPR gives uncertainty estimates (helpful when predicting outside grid).
    - NN useful to compare but needs care (regularization, early stopping) given limited data.
3. **Uncertainty / reliability:** recommend GPR or Bayesian/quantile methods for uncertainty estimation or bootstrap ensembles for prediction intervals.

**Training protocol (required)**

1. **Data split:** Use k-fold cross-validation (recommended k=5 or 10) for model selection. Additionally, keep a held-out test set (e.g., 15–20%) for final evaluation.
2. **Hyperparameter tuning:** use grid search or Bayesian optimization (MATLAB bayesopt or hyperparameters options) inside CV.
3. **Evaluation metrics:** report **RMSE**, **MAE**, and $\mathbf{R^2}$ on CV and held-out test set. Also report prediction intervals where feasible.
4. **Residual analysis:** plot residuals vs predicted, residual histogram, Q-Q plot.
5. **Model interpretability:** produce feature importance (for tree models) and partial dependence plots (PDP). Optionally compute SHAP values (if using external toolboxes).

**(Students can use either script or MATLAB's Regression Learner App; both are acceptable, include exact commands if using code.)**

**Required plots and outputs (deliverables within report)**

Students must include:

1. Histogram of Deformation.
2. Parity (predicted vs actual) for top models (with RMSE, MAE, $R^2$ annotated).
3. Residual vs predicted scatter and residual histogram.
4. Feature importance bar chart (for tree models).
5. Partial dependence plots for two most important features (2D surface or 1D PDP).
6. CV learning curves or training/validation error vs model complexity/hyperparameter.
7. If GPR is used: predicted mean ± 2×SD plotted for test points (shows predictive uncertainty).
8. (Optional but encouraged) 2D heatmap slices: fix two inputs, vary the other two across a fine grid and plot predicted deformation to visualize surrogate surface.

**Technical report: required structure & what to write**

Students must submit a PDF technical report (max 5 pages, excluding figures & references). Use the sections below.

**Suggested report sections**

1. **Title and authors**
2. **Abstract** (≤200 words)
3. **Introduction**
   - Brief description of the actuator and why surrogate modeling is useful.
   - Define inputs and outputs and scope of assignment.
4. **Problem statement**
   - Data source (ANSYS simulations), grid description
5. **Data description**
   - Table of parameters (min/max/interval) and dataset statistics (counts, missing entries).
6. **Methodology** ← *Complete example text provided below*
7. **Results** ← *Complete example text provided below*
8. **Discussion**
   - Interpret model performance, limitations, uncertainty, reliability in extrapolation.
9. **Test case(s)** ← *Required: at least 2 unseen input combos including an "in-between" point outside original grid and one near the training domain edge*
10. **Conclusions and recommendations**
11. **Appendix**

- Full code listing, hyperparameter tables, additional plots, raw metric tables.

12. **References**

**Test cases (explicit)**

Students must include at least **two** test cases where the model predicts deformation and provide reasoning about model confidence:

1. **Interpolative test** (in-between grid values): e.g., Pressure=1250, Height=5.25, Length=55, Thickness=1.05.
2. **Edge test** (on grid boundary): e.g., Pressure=5000, Height=8.0, Length=100, Thickness=1.2.

For each test case report predicted value, uncertainty (if available), and recommended next action (e.g., “if predicted uncertainty > X then run FEA to validate”).

## Deliverables (students must submit)

1. **MATLAB code** (script(s) + functions) that:
    - loads and cleans the dataset,
    - trains the candidate models,
    - performs hyperparameter tuning and cross-validation,
    - saves final model(s),
    - produces required plots (parity, residuals, PDPs, histograms).
    - Code must be runnable and documented (README with MATLAB version).

2. **Technical report** (PDF) following the structure above (including code appendix).

# Grading Rubric (100 points total)

### 1. Introduction & Problem Statement (5 pts)

- Clearly introduces the actuator and the need for a surrogate model.
- Defines the four input parameters and the single output (Deformation).
- States the overall objective of the assignment.

### 2. Data Description & Exploration (5 pts)

- Presents a table of input parameters with their ranges (min, max, interval).
- Includes a correctly labeled histogram of the 'Deformation' output variable.
- Reports basic dataset statistics (e.g., number of samples).

### 3. Methodology (25 pts)

- Details the train/test data split methodology and percentages used.
- Names the specific regression models implemented (at least two).
- Describes the k-fold cross-validation process for model selection.
- Explains the hyperparameter tuning approach (e.g., Bayesian optimization, grid search).
- Lists the specific evaluation metrics used (RMSE, MAE, $R^2$).

### 4. Results & Visualization (30 pts)

- **Parity Plots:** Predicted vs. Actual for final models, annotated with $R^2$, RMSE, and MAE.
- **Residual Analysis:** Residual vs. Predicted plot and a histogram of residuals.
- **Feature Importance:** Bar chart for the best tree-based model.
- **Partial Dependence Plots (PDPs):** Plots for the two most important features.
- **Uncertainty Plot (if GPR used):** Predicted mean ± 2*SD for test points.
- **Clarity:** All plots must be high-quality, clearly labeled, and titled.

### 5. Discussion & Interpretation (25 pts)

- Compares model performance and justifies the selection of the best model.
- Interprets feature importance and PDPs in the context of the actuator's behavior.
- Discusses model limitations, reliability, and challenges with extrapolation.

### 6. Test Cases & Recommendations (10 pts)

- Reports the predicted deformation for the required "interpolative" test case.
- Reports the predicted deformation for the required "edge" test case.
- Assesses model confidence for each prediction and provides an actionable recommendation (e.g., "run FEA to validate").

# Academic Integrity and Originality Policy

This technical report is an individual assessment of your analytical and communication skills. All submissions must represent your own original work and adhere to the university's academic integrity standards.

Your report will be automatically processed by **Turnitin** upon submission to evaluate two key metrics: textual similarity and the use of AI-generated content.

**Submission Requirements:**

- **Similarity Index:** Your report must not exceed a **15%** similarity score. This allowance accounts for properly cited sources, standard technical phrasing, and assignment template language.
- **AI Writing Indicator:** The AI-generated content score must not exceed **10%**. You may use AI tools for brainstorming, code debugging, or checking grammar, but the report's narrative, analysis, and conclusions must be written entirely by you.

**Penalties for Non-Compliance:**

The following penalties are strict and will be applied automatically:

- A score exceeding the **15% Similarity** or **10% AI** limit will result in an automatic **50% deduction** from your total report mark.
- A submission with over **30% Similarity** and **30% AI-generated content** will be considered a severe violation and will receive a **grade of zero (0)**.

**Pre-Submission Check:**

You have the ability to submit your report to Turnitin before the final deadline to check your scores. It is your responsibility to review your report and make any necessary revisions to meet these requirements prior to final submission.