Automated Gear Design: A Parametric Approach Using Python

Team Members

1. Rithvik Subash - 22B2120
2. Shahu Patil - 22B2146
3. Ayush Singh - 22B2203

1. Introduction

Effective mechanical systems rely on well-designed gears. This project aims to provide a Python application for automated bevel gear design. To develop efficient bevel gear designs, the computer will take into account gear specifications (module, number of teeth, pressure angle) as well as design parameters (shaft angles, gear ratio). The outputs will contain complete gear profiles, tooth dimensions, and, if necessary, 3D models for display.

This program will improve the design process by lowering errors and design time when compared to manual techniques. It will be a helpful tool for engineers and designers, allowing them to swiftly create precise and customizable bevel gear designs for a variety of mechanical applications.

2. Objectives

1. Create a Python program that accepts user input for gear characteristics such as module, number of teeth, and pressure angle.
2. Add design characteristics to the program, such as shaft angles and gear ratio, for more customised gear designs.
3. Use algorithms to determine optimal gear profiles and tooth size depending on input factors.
4. Generate exact gear profiles and tooth measurements for manufacturing needs.
5. Optionally, incorporate the ability to create 3D models of the designed bevel gears for display and analysis.
6. Ensure the program's user interface is intuitive, allowing for simple parameter entry and result interpretation.
7. Run the program through several input scenarios to ensure accuracy and dependability in creating bevel gear designs.
8. Provide documentation and instructions for the program, including examples and troubleshooting advice.

3. Procedure

1. Research and Analysis:

Conduct a literature review on bevel gear design principles and algorithms.
Analyze existing software tools for automated gear design to identify best practices.

2. Programming Environment Setup:

Set up Python development environment with necessary libraries for mathematical calculations and 3D modeling (if applicable).

3. User Input Handling:

Develop functions to handle user inputs for gear specifications and design parameters.

Validate user inputs to ensure they fall within acceptable ranges.

4. Algorithm Implementation:

- Implement algorithms to calculate gear profiles, tooth dimensions, and other critical parameters based on user inputs.
- Ensure the algorithms are efficient and capable of handling a wide range of input scenarios.

5. Output Generation:

- Develop functions to generate detailed gear profiles and tooth dimensions based on the calculated parameters.
- Optionally, implement functionality to generate 3D models of the designed bevel gears.

6.Testing and Validation:

- Test the program with various input scenarios to validate its accuracy and reliability.
- Make any necessary adjustments to the algorithms or user interface based on testing feedback.

7. Documentation and Deployment:

- Prepare detailed documentation for the program, including instructions for use, examples, and troubleshooting tips.
- Deploy the program for use by engineers and designers, ensuring it meets their needs for automated bevel gear design.

4. Timeline

1. March 10 - March 15:

- Research and analyze bevel gear design principles and existing software tools.
- Set up Python development environment.

2. March 16 - March 20:

- Develop user input handling functions.
- Begin implementing algorithms for gear profile and tooth dimension calculations.

3. March 21 - March 25:

- Continue algorithm implementation.

- Start developing output generation functions.

4. March 26 - March 30:
- Complete algorithm implementation.
- Test algorithms with sample inputs.

5. March 31 - April 5:
- Integrate user input handling, algorithm, and output generation functions into the user interface.

6. April 6 - April 8:
- Test the program with various input scenarios.
- Make any necessary adjustments based on testing feedback.

7. April 9 - April 10:
- Finalize documentation, including instructions for use, examples, and troubleshooting tips.
- Deploy the program for use by engineers and designers.