An LLM generated shaft design script for Fusion 360

Dr. Gökçe Mehmet AY

July 19, 2024

1 Introduction

Large Language Models (LLMs) have demonstrated proficiency in various domains. While specialized machine learning algorithms exist for mechanical design, the potential of general-purpose LLMs like Google Gemini in this field remains underexplored. This study investigates the feasibility of utilizing such an LLM to develop a simplified script for shaft design, given user-specified loads and distances. For this initial exploration, material properties were held constant. The generated shaft design is then translated into a CAD file using the Fusion 360 API in Python. The procedural flowchart is detailed in Figure 1, with corresponding code snippets and explanations provided in Section 2. The script can be found on Github Page.

2 The Code Snippets and Explanations

2.1 Get Load and Distance

```
loadInput, _ = ui.inputBox('Enter the load (Newtons):', 'Shaft Design')
distanceInput, _ = ui.inputBox('Enter the distance between supports (mm):', 'Shaft Design')
load = float(loadInput)
distance = float(distanceInput) / 10.0 # Convert to cm
```

Prompts the user for the load (F) in Newtons (N) and the distance between supports (L) in millimeters (mm). The distance is then converted from millimeters to centimeters for consistency with Fusion 360's internal unit system.

2.2 Calculate Shaft Diameter

```
maxBendingMoment = load * distance / 4
allowableStress = yieldStrength / safetyFactor
shaftDiameter = ((32 * maxBendingMoment) / (math.pi *
    allowableStress)) ** (1/3)
```

Calculates the minimum required shaft diameter (d) based on the applied load, material properties, and a safety factor. The following formulas are used:

Maximum Bending Moment (M_{max}):

$$M_{max} = \frac{F \cdot L}{4}$$

Allowable Stress (σ_{allow}):

$$\sigma_{allow} = \frac{\sigma_{yield}}{n}$$

Shaft Diameter (d):

$$d = \sqrt[3]{\frac{32 \cdot M_{max}}{\pi \cdot \sigma_{allow}}}$$

2.3 Create Shaft Geometry

```
newOcc = rootComp.occurrences.addNewComponent(adsk.
    core.Matrix3D.create())
newComp = adsk.fusion.Component.cast(newOcc.component)
sketches = newComp.sketches
xyPlane = newComp.xYConstructionPlane
sketch = sketches.add(xyPlane)
# ... (create circle and extrude profile to create
    shaft)
```

Creates a new component and then adds a sketch to it. Within the sketch, a circle is created at the origin with the calculated diameter. The circle is extruded to the specified length to form the shaft.

2.4 Add Dimensions

```
# ... (get SketchPoints and create start/end points)

 shaftDiameterDimension = mainBodyDimensions.
    addDiameterDimension(circle, adsk.core.Point3D.
    create(shaftDiameter / 2 + 10, 0, distance / 2))
            # Get SketchPoints collection
            sketchPoints = mainBodySketch.sketchPoints
```

```
6
7               # Create Point3D objects from BRepVertex
                    objects
8               startVertex = newComp.bRepBodies.item(0).
                    edges[0].startVertex
9               endVertex = newComp.bRepBodies.item(0).
                    edges[0].endVertex
10              startPoint = sketchPoints.add(startVertex.
                    geometry)
11              endPoint = sketchPoints.add(endPoint.
                    geometry)
12
13              shaftLengthDimension = mainBodyDimensions.
                    addDistanceDimension(
14                   startPoint,
15                   endPoint,
16                   adsk.fusion.DimensionOrientations.
                         AlignedDimensionOrientation,
17                   adsk.core.Point3D.create(0, 0,
                         distance / 2)
```

Adds dimensions to the shaft model, specifically the length and diameter.

2.5 Display Results

```
1 ui.messageBox(f"Shaft Diameter: {shaftDiameter:.2f} mm
      \nShaft Length: {distance * 10:.2f} mm\n") #
      Convert back to mm for display
```

Displays the calculated shaft diameter (d) and length (L) in a message box to the user. The length is converted back to millimeters for display.

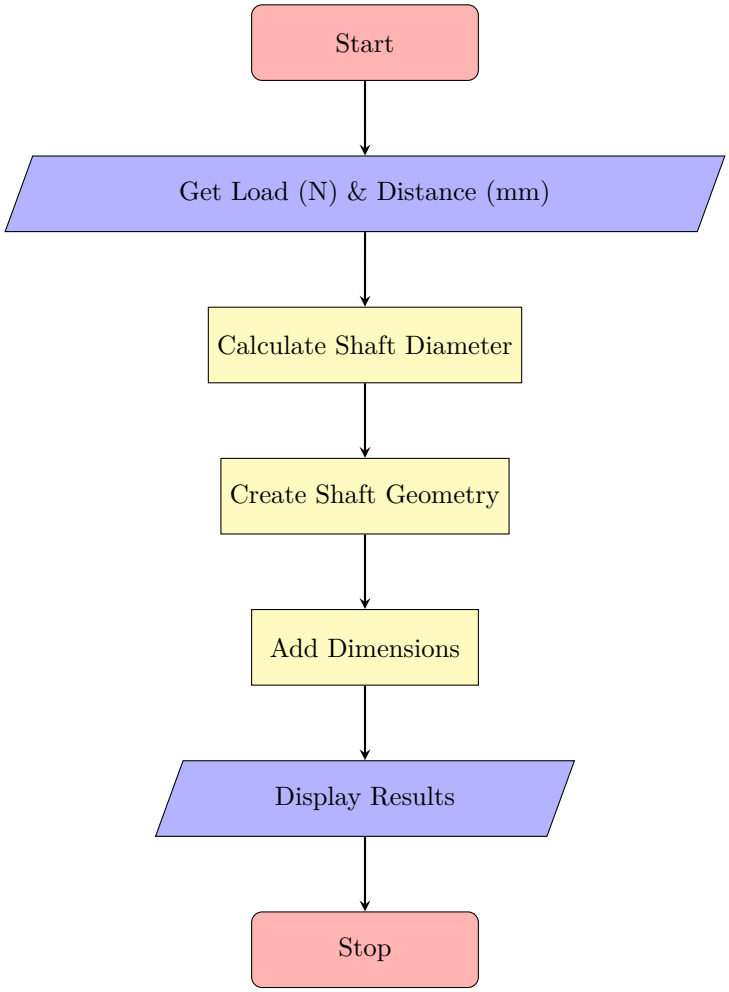

Figure 1: Shaft Design Flowchart