Composites Filament Winder Documentation

Original Code: Rylan Andrews, Dalton Reichenbach

Troubleshooting: Evan Mullins

1 Composites Filament Winder Project Overview

The project contains code and documentation for the filament winder.

1.1 Project Structure

The project is organized into two main folders:

1. **generator**: Program used on a desktop/laptop to create a configuration file that is transferred to the Arduino. This is where wind angle, plies, and other settings are configured.
2. **documentation**: Contains documentation for the winder (currently empty).

1.2 Python Files in Generator

There are seven Python files with varying functionality:

- `definitions.py` — Internal definitions for the rest of the code.
- `gui.py` — GUI version of `main.py` using Tkinter; primary interface for G-code generation.
- `helper.py` — CLI helpers: printing help, loading schedules, generating G-code, and calculations.
- `load.py` — Loads wind schedule data from a JSON file.
- `main.py` — Original CLI program (does not set default feed rate).
- `planner.py` — Computes internal math to plan the filament winding paths.
- `winder.py` — Simulates the winder and generates G-code commands.

Only two files are directly used to output G-code:

- `gui.py` — Updated GUI version.
- `main.py` — Original CLI version (does not automatically add feed rate line).

1.3 Key Helper Files

definitions.py Defines wind properties, including:

- **Wind angle** — Angle between mandrel axis and tow direction.
- **Number of starts** — Number of start positions per pattern.
- **Lock angle** — Angle through which mandrel turns at the end of each pass.

helper.py Provides CLI functions:

- `help` — Print command menu.
- `load` — Load wind schedule.
- `generate` — Generate G-code.
- `calculator` — Compute valid wind parameters.

load.py Loads JSON configuration files into the program.

planner.py Contains all mathematical calculations for winding paths.

winder.py Simulates the physical winder and outputs G-code.

Axis definitions:

- X-axis: carriage motion along mandrel (inches)
- Z-axis: mandrel rotation (degrees)
- Y-axis: currently unused

Key G-code outputs:

- `G20` — Set units to inches
- `G01 F100.0` — Feed rate (missing in `main.py`)
- `G92 X0 Z0` — Set axis position
- `G28` — Move home (currently not called)

1.4 Issue: Feed Rate Line Missing in CLI

The F-line (`G01 F...`) is not generated in `main.py` because:

- `Winder` constructor sets `currentFeedRate` to the default.
- `setFeedRate` only appends a line if feed rate changes or `force=True`.
- CLI did not call `setFeedRate` with `force=True`.

Solution:

1. Update `Winder.setFeedRate`:

```
def setFeedRate(self, f, force=False):
    if force or self.currentFeedRate is None or f != self.currentFeedRate:
        self.gcode.append(f"G01 F{round(f,3)}")
        self.currentFeedRate = f
```

2. Remove feed rate from constructor or set `currentFeedRate=None`.

3. Always call `setFeedRate(..., force=True)` before generating G-code:

   ```
   machine = winder.Winder()
   machine.setFeedRate(defaultFeedRate, force=True)
   gcode = planner.planWind(schedule, machine)
   ```

4. Optionally call `machine.moveHome()` to add homing line (`G28`).