Code Documentation

❖ Overview

This MATLAB script is designed to visualize in real-time the behavior of the solar tracking system implemented using an Arduino-based dual-mode controller (PID & non-PID). The goal of this script is to monitor key performance indicators such as:

- Light intensity readings from LDR sensors (LDR1 & LDR2)
- Servo motor angle (representing panel position)
- Current mode of operation (Automatic vs Manual)

❖ Connection to Arduino System

The MATLAB script receives data via the serial port from the Arduino microcontroller.

The Arduino code sends the data through Serial.print() every 20 milliseconds in a comma-separated format, as follows:

```
<time_ms>,<ldr1>,<ldr2>,<angle>,<pid_output>,<mode>
```

Example line from serial output:

```
12560,43,59,22.5,1.84,1
```

Note:

This MATLAB script will not function unless the Arduino system is connected via USB, and the correct COM port and baud rate (115200) are set in the script:

```
port = "COM6";      % Change this to match your system

baud = 115200;
```

If no Arduino is connected, the script will throw a connection error or wait indefinitely for data.

For best results, make sure:

- The Arduino code is already uploaded and running.
- The Serial Monitor on the Arduino IDE is closed (to avoid port conflict).
- The hardware is powered and functional.

❖ What the Code Does

1. Initial Setup

- Connects to the specified COM port.
- Initializes data buffers and a live figure window with 3 subplots:
- LDR Readings (0–100%)
- Panel Angle (−90° to +90°)
- Operating Mode (Auto / Manual)

2. Live Data Loop

- Continuously reads and parses the serial data stream.
- Buffers the last 1000 points for plotting.
- Updates the plots every 100 ms.

3. Data Filtering

- Automatically skips any malformed or incomplete lines.
- Uses circular buffering to keep memory usage efficient.

❖ Advantages of This Visualization

- Allows real-time observation of how the system responds to changing light conditions.
- Helps compare the performance between PID-controlled and non-PID-controlled behavior.
- Provides visual evidence of oscillations, stabilization time, and accuracy.

❖ Use Case

This script is primarily used to evaluate and present the real-time response of the solar tracking system during:

- Lab testing and debugging
- Performance comparison
- Academic present