

This code creates a graphical user interface (GUI) using the **Dear PyGui** library to control an ODrive motor, monitor its position and applied force, and perform sinusoidal force modulation based on user inputs.

Code Structure Overview:

1. Motor Initialization:

- The ODrive motor is located at the beginning using **odrive.find_any()**.
- Functions for motor control, such as calibration, turning on/off, setting force, and moving to a specific position, are defined as follows:
 - **calibrate()**: Calibrates the motor.
 - **turn_on()**: Toggles the motor's state between on (closed-loop control) and off.
 - **set_force_kg()**: Sets the force applied by the motor in kilograms.
 - **move_to(position)**: Moves the motor to a specified position.

2. SineModulation Class:

- This class defines a sine wave function used to modulate values, such as the applied force, over time.
- It takes two main parameters:
 - **Frequency**: Determines how fast the sine wave oscillates.
 - **Amplitude**: Controls the range of the oscillation.
- The **modulate()** method returns the dynamically modulated set point based on time, enabling smooth force adjustments.
 - The function to modulate the set point based on position is not yet implemented.

3. Graph and Data Logging:

- The script records and logs motor data, such as position and applied force, which are plotted in real-time.
- The graph is updated dynamically using **Dear PyGui's** plotting features.
- **Graph Update Rate**: Set to 60 Hz (**graph_update_rate = 1.0 / 60.0**), ensuring smooth updates on the plotted data. Worth noting is that this can't be higher than the screen's update ratio.

- When data logging is enabled, it records the position, force, and time into a CSV file for later analysis.

4. User Interface (UI):

The GUI allows the user to interact with the motor control system through several components:

- **Buttons:**
 - Controls for motor calibration, turning the motor on/off, clearing errors, and setting weight.
- **Input Fields:**
 - Users can input their desired force (in kilograms), and configure sine wave modulation settings such as:
 - Frequency
 - Minimum and maximum force.
- **Modulation Options:**
 - Checkboxes allow users to toggle between time-based or position-based sine wave modulation of force.

5. Main Loop:

- The primary loop continuously checks if the GUI is running and performs several operations:
 - Updates the graphs with the motor's current position and applied force.
 - If logging is active, it records data into a CSV file.
 - Applies the sine modulation when the respective checkbox (time-based) is checked.

6. Key Functions:

- **Motor Control:**
 - **calibrate()**: Calibrates the ODrive motor.
 - **turn_on()**: Toggles between enabling and disabling the motor.
 - **set_force_kg(kg)**: Sets the motor force to the specified value in kilograms.
 - **move_to(position)**: Moves the motor to the desired position.
 - **get_current_position()**: Retrieves and returns the motor's current position.
- **Graph and Data Updates:**

- **update_graphs()**: Updates the plots with the latest motor position and applied force data.
- **start_recording()**: Starts or stops the data logging process.
- **move_increment()**: Allows the user to move the motor by a specified increment (e.g., for fine adjustments).

7. How it Works:

- The GUI offers real-time feedback, displaying the motor's position and the force being applied.
- Users can manually input desired weights to apply to the motor or enable sine wave modulation to automatically adjust the applied force over time.
- Data (position, force, and time) is logged and dynamically plotted in real-time. This is managed efficiently by the main event loop, ensuring continuous operation while applying user-defined motor controls.

In summary, this script integrates motor control, data logging, and real-time feedback through a GUI. It enables users to adjust the force applied by an ODrive motor and visualize the effects through dynamic graphs. The sine modulation feature adds flexibility in controlling the motor's behavior.