**Bài tập Coding: Đồ thị Mô phỏng Oscilloscope**

**Giới thiệu**

Ứng dụng này là một mô phỏng đơn giản của máy hiện sóng (oscilloscope), hiển thị một tín hiệu đầu vào duy nhất. Giao diện gồm ba đồ thị, mỗi cái minh họa một khía cạnh khác nhau của hoạt động oscilloscope. Người dùng có thể điều chỉnh các thông số của tín hiệu đầu vào như biên độ, tần số, pha, loại hàm (sin hoặc cos), cũng như cách tín hiệu quét hoạt động thông qua hệ số quét.

**Hàm update\_plot**

Hàm update\_plot là trung tâm của toàn bộ mô phỏng. Mỗi khi người dùng thay đổi một tham số, hàm này sẽ:

1. **Đọc đầu vào và thiết lập tham số:** Lấy giá trị hiện tại từ các điều khiển (thanh trượt, nút radio...).
2. **Tính toán tín hiệu và tín hiệu quét:** Dựa trên lựa chọn sin/cos và các tham số hiện tại, hàm tính y(t) và tín hiệu quét x\_sweep\_signal(t) – là một dạng sóng răng cưa. Pha quét thuận và quét ngược được tính dựa trên chu kỳ của tín hiệu đầu vào.
3. **Vẽ đồ thị:** Cập nhật dữ liệu lên ba trục đồ thị Matplotlib. Có cờ is\_plot\_initialized để phân biệt lần vẽ đầu tiên và các lần cập nhật sau, giúp tiết kiệm tài nguyên.

**Chi tiết các Đồ thị**

**Đồ thị 1: Tín hiệu đầu vào theo thời gian**

* **Mục đích:** Hiển thị tín hiệu đầu vào y(t) theo thời gian.
* **Cách thực hiện:** Cập nhật dữ liệu line\_input bằng set\_data(). Có thêm chú thích hiển thị thông số tín hiệu. Giới hạn trục tự động điều chỉnh theo thời gian và biên độ.

**Đồ thị 2: Tín hiệu quét theo thời gian**

* **Mục đích:** Trực quan hóa tín hiệu quét bên trong oscilloscope – dạng sóng răng cưa.
* **Cách thực hiện:** Vẽ tín hiệu quét x\_sweep\_signal(t). Pha quét thuận (0 → 1) và quét ngược (1 → 0) được chú thích rõ bằng văn bản. Trục Y cố định từ 0 đến 1, trục X điều chỉnh theo thời gian.

**Đồ thị 3: Tín hiệu Y so với vị trí quét**

* **Mục đích:** Mô phỏng vệt sáng trên màn hình oscilloscope khi tín hiệu Y biến đổi theo quét X.
* **Cách thực hiện:** Mỗi lần cập nhật, trục này được xóa và vẽ lại toàn bộ. Với mỗi chu kỳ, phần quét thuận được vẽ bằng đường liền nét (màu lục lam), phần quét ngược vẽ bằng đường đứt nét (màu đỏ hoặc hồng). Có chú thích giúp phân biệt rõ hai phần này.

**Cập nhật giao diện**

Cuối cùng, fig.tight\_layout() điều chỉnh bố cục để không bị chồng chữ, và canvas.draw() sẽ hiển thị toàn bộ nội dung cập nhật lên giao diện.

**Các Tính năng khác:**

* **Giao diện:** Xây dựng bằng Tkinter (tk, ttk).
* **Đồ thị:** Sử dụng Matplotlib, tích hợp trực tiếp vào cửa sổ Tkinter.
* **Bố cục:** Dùng PanedWindow để dễ thay đổi kích thước các phần.
* **Điều khiển:** Thanh trượt (ttk.Scale), ô nhập số (ttk.Entry), nút radio (chọn đơn vị pha, loại sóng), combobox (chọn theme), nút bấm (chỉnh pha nhanh).
* **Xác thực:** Ô nhập chỉ nhận số khi nhấn Enter.
* **Theme:** Có thể đổi giữa các chủ đề giao diện.
* **Button Warping:** Nút “Quick Set” tự động xuống dòng nếu cửa sổ bị thu hẹp.
* **Chỉnh sửa cỡ chứ**: từ 10 - 18
* **Khởi tạo:** Hàm final\_initialization() khởi tạo toàn bộ và bắt đầu vòng lặp root.mainloop().