

Smoothing Code Explanation – For the Teacher

Smoothing data: Smoothing techniques are important for reducing noise in a signal, making it easier to identify meaningful patterns or trends. In real-world data such as EMG signals, we often encounter fluctuations and random variations, which can obscure the underlying signal. By applying smoothing methods, we can eliminate or reduce these fluctuations, making the data cleaner and more interpretable.

Smoothing Script Explanation

The following is a Python code snippet that performs smoothing on downsampled audio or signal data using the Savitzky-Golay filter, a common signal processing technique.

Python Script for Smoothing (In the Jupyter Notebook)

```
# Apply smoothing
smoothed_data = savgol_filter(downsampled_data, 51, 3) # Apply Savitzky-
Golay filter to smooth the data, with a window size of 51 and polynomial
order 3
df_smoothed = pd.DataFrame({'Time (seconds)': time_axis_downsampled,
'Amplitude': smoothed_data}) # Create a DataFrame with smoothed time and
amplitude
df_smoothed.to_csv('/content/drive/MyDrive/Colab
Notebooks/outputFile_smoothed.csv', index=False) # Save the smoothed
DataFrame as a CSV file
```

Explanation of the Script:

```
smoothed_data = savgol_filter(downsampled_data, 51, 3) # Apply Savitzky-
Golay filter to smooth the data, with a window size of 51 and polynomial
order 3
```

- **Explanation:** This line applies the **Savitzky-Golay filter**, which is a smoothing technique, to the `downsampled_data`.
 - **savgol_filter():** This is a function from the **scipy.signal** module that applies the Savitzky-Golay filter, a widely used technique for smoothing data. The filter works by fitting successive polynomials to a moving window of data points.
 - **Parameters:**
 - **downsampled_data:** The raw data that needs to be smoothed (the signal after downsampling).
 - **51:** The **window size** (number of points used to compute each smoothed value). A window of 51 means the filter uses 51 data points to compute each new value. It is an odd number, as the center point of the window is used as the "current" value.
 - **3:** The **polynomial order** used for the smoothing. In this case, it uses a 3rd-degree polynomial. This means the filter will fit cubic polynomials to the data in each window.

Name:

Date:

Class:

- **Result:** The `savgol_filter()` applies the smoothing and returns a new array of smoothed values, which is assigned to the variable `smoothed_data`.

Teacher's Guide:

"The Savitzky-Golay filter is a powerful method to smooth data. Here, we've chosen a window size of 51 and a polynomial order of 3. This means that for each data point, the algorithm fits a cubic polynomial to 51 surrounding points and calculates a smoothed value. This helps reduce noise while maintaining the shape of the signal."

```
df_smoothed = pd.DataFrame({'Time (seconds)': time_axis_downsampled,
                             'Amplitude': smoothed_data}) # Create a DataFrame with smoothed time and
amplitude
```

- **Explanation:** This line creates a **Pandas DataFrame** that stores the smoothed data.
 - **pd.DataFrame():** This function is used to create a DataFrame, which is a table-like structure for handling and analyzing data.
 - **{'Time (seconds)': time_axis_downsampled, 'Amplitude': smoothed_data}:** The DataFrame is created using a dictionary. The dictionary has two keys:
 - **'Time (seconds)':** The column name for the time values associated with the downsampled signal.
 - **time_axis_downsampled:** This is the time axis for the downsampled signal that was created earlier in the code (it matches the length of the downsampled_data).
 - **'Amplitude':** The column name for the amplitude (or signal strength) values.
 - **smoothed_data:** The smoothed signal values that were calculated using the Savitzky-Golay filter.
 - **Result:** The DataFrame `df_smoothed` now contains two columns: one for the **time** values and another for the **smoothed amplitude** values.

Teacher's Guide:

"Here, we're creating a table of the smoothed signal. The first column contains the time values, and the second column holds the amplitude values of the smoothed signal. This structure allows us to easily analyze the data or save it for further use."

```
df_smoothed.to_csv('/content/drive/MyDrive/Colab
Notebooks/outputFile_smoothed.csv', index=False) # Save the smoothed
DataFrame as a CSV file
```

- **Explanation:** This line saves the smoothed DataFrame (`df_smoothed`) as a .csv file.
 - **df_smoothed.to_csv():** This function saves the DataFrame `df_smoothed` to a .csv file.
 - **'/content/drive/MyDrive/Colab Notebooks/outputFile_smoothed.csv':** This is the path where the .csv file will be saved. In this case, it's being saved to Google Drive in the folder Colab Notebooks. The file will be named `outputFile_smoothed.csv`.

Name:

Date:

Class:

- **index=False:** This argument prevents Pandas from saving the index (row numbers) as an additional column in the .csv file. This keeps the .csv file clean and focused only on the data columns (time and amplitude).
- **Result:** The smoothed data is saved as a .csv file in the specified path, making it available for further analysis, sharing, or exporting.

Teacher's Guide:

"Once the data is smoothed, we're saving it as a .csv file. This makes it easy for you to share or load the data into other programs like Excel for further analysis."

Summary of the Process:

1. **Smoothing the Data:** The Savitzky-Golay filter is used to smooth the downsampled signal by applying a polynomial smoothing technique with a specified window size and polynomial order.
2. **Creating a DataFrame:** The time values and smoothed amplitude values are stored together in a Pandas DataFrame, making the data organized and easy to handle.
3. **Saving to CSV:** The smoothed data is saved as a .csv file, allowing students to access and further process the data outside of Python.

Patterns in Smoothed Data

- **What students will see:**
 - **Clearer Trends:** The smoothed data will show the main peaks and troughs of the signal, representing periods of high and low muscle activity.
 - **Reduced Noise:** Random fluctuations in the data will be minimized, making it easier to focus on the overall signal behavior.
 - **Consistency:** Patterns such as repetitive bursts of activity or steady oscillations will become more apparent.
- **Relation to muscle activity and movement:**
 - Peaks in the smoothed signal correspond to periods of **muscle contraction** or activity.
 - Troughs represent **relaxation** or minimal activity.
 - The timing and magnitude of these patterns can indicate the strength and frequency of muscle activity.