

PRETIME Tutorial

Understanding MATLAB Code for EEG Signal Processing

matlab code eeg signal is a powerful combination that enables researchers, engineers, and neuroscientists to analyze and interpret electroencephalogram (EEG) data effectively. EEG signals are critical in understanding brain activity, diagnosing neurological conditions, and developing brain-computer interface (BCI) systems. MATLAB, with its extensive toolboxes and user-friendly scripting environment, provides a comprehensive platform to process EEG signals, extract meaningful features, and visualize results. This article explores how MATLAB code can be used for EEG signal processing, including data acquisition, filtering, feature extraction, and visualization techniques.

Introduction to EEG Signals

Electroencephalogram (EEG) signals are electrical activities recorded from the scalp, representing the collective neural oscillations in the brain. These signals are characterized by their amplitude, frequency, and phase, with different patterns associated with various mental states, tasks, or neurological conditions. EEG signals are typically sampled at rates between 128 Hz and 1024 Hz, depending on the application.

Key features of EEG signals include:

- Frequency bands: Delta (0.5-4 Hz), Theta (4-8 Hz), Alpha (8-13 Hz), Beta (13-30 Hz), Gamma (>30 Hz)
- Amplitude variations: Ranging from a few microvolts to hundreds of microvolts
- Artifacts: Noise from muscle activity, eye movements, and external electrical interference

Effective analysis requires preprocessing steps to clean and prepare the data for further analysis.

Getting Started with MATLAB for EEG Signal Processing

MATLAB offers dedicated toolboxes such as the Signal Processing Toolbox and the EEGLAB toolbox, which facilitate EEG data analysis. To begin, you need to load your EEG data into MATLAB, which can be in formats like .mat, .edf, .bdf, or plain text files.

Loading EEG Data

```
```matlab
```

```
% Example: Load EEG data stored in a MATLAB file
```

```
EEG = load('eeg_data.mat');
```

```
% Assuming the data is stored in EEG.data
```

```
signal = EEG.data;
```

```
samplingRate = EEG.samplingRate;
```

```
```
```

Visualizing Raw EEG Data

```
```matlab
```

```
timeVector = (0:length(signal)-1) / samplingRate;
```

```
figure;
```

```
plot(timeVector, signal);
```

```
xlabel('Time (s)');
```

```
ylabel('Amplitude (\muV)');
```

```
title('Raw EEG Signal');
```

```
```
```

Preprocessing Steps

- Filtering: Remove noise and artifacts.
- Artifact Removal: Detect and eliminate eye blinks or muscle activity.
- Segmentation: Divide signals into epochs for task-specific analysis.

Filtering EEG Signals Using MATLAB

Filtering is essential to isolate specific frequency bands or remove unwanted noise. MATLAB provides functions like ``bandpass``, ``lowpass``, and ``highpass`` for filtering purposes.

Designing Filters

Let's design a bandpass filter to isolate alpha waves (8-13 Hz):

```
```matlab

% Define filter parameters

lowCutoff = 8; % Hz

highCutoff = 13; % Hz

filterOrder = 4;

% Design Butterworth bandpass filter

[b, a] = butter(filterOrder, [lowCutoff, highCutoff]/(samplingRate/2), 'bandpass');

% Apply filter

alphaEEG = filtfilt(b, a, signal);

```
```

Visualize Filtered Signal

```
```matlab

figure;

plot(timeVector, signal, 'b', 'DisplayName', 'Raw Signal');

hold on;

plot(timeVector, alphaEEG, 'r', 'DisplayName', 'Alpha Band');

xlabel('Time (s)');
```

```
ylabel('Amplitude (\muV)');

title('EEG Signal Before and After Filtering');

legend();

hold off;

````
```

Artifact Detection and Removal in EEG Data

Artifacts such as eye blinks and muscle movements can distort EEG analysis. MATLAB code can be used to detect these artifacts based on amplitude thresholds, statistical properties, or Independent Component Analysis (ICA).

Simple Artifact Detection Using Thresholding

```
``matlab  
  
% Define amplitude threshold (e.g., 100  $\mu$ V)  
  
threshold = 100;  
  
% Find segments exceeding threshold  
  
artifactIndices = find(abs(alphaEEG) > threshold);  
  
% Mark artifacts  
  
artifactMask = false(size(alphaEEG));  
  
artifactMask(artifactIndices) = true;  
  
% Remove artifacts by interpolation  
  
cleanEEG = alphaEEG;  
  
cleanEEG(artifactMask) = interp1(timeVector(~artifactMask), alphaEEG(~artifactMask),  
timeVector(artifactMask), 'linear');
```

...

Using ICA for Artifact Removal

The EEGLAB toolbox simplifies ICA implementation:

```
```matlab

% Assuming EEG data is loaded into EEGLAB structure

EEG = pop_importdata('data', 'path_to_data');

EEG = pop_runica(EEG, 'extended', 1);

% Visualize components and remove artifacts

pop_topoplot(EEG, 0, [1:size(EEG.icaweights,1)], 'IC scalp maps');

% Remove artifact-related components manually or automatically

EEG_clean = pop_subcomp(EEG, [components], 0);

...

```

## Feature Extraction from EEG Signals

Once the EEG data is filtered and cleaned, extracting features such as band power, entropy, or wavelet coefficients is critical for classification, diagnosis, or BCI applications.

### Power Spectral Density (PSD)

Estimating the power in different frequency bands helps understand brain activity patterns.

```
```matlab

% Use Welch's method

windowSize = 2 * samplingRate; % 2 seconds window

[pxx, f] = pwelch(cleanEEG, windowSize, windowSize/2, [], samplingRate);

```

% Plot PSD

figure;

plot(f,10log10(pxx));

xlabel('Frequency (Hz)');

ylabel('Power/Frequency (dB/Hz)');

title('EEG Power Spectral Density');

xlim([0 50]);

...

Calculating Band Power

```matlab

% Define frequency bands

bands = [0.5 4; 4 8; 8 13; 13 30; 30 50];

bandNames = {'Delta','Theta','Alpha','Beta','Gamma'};

bandPower = zeros(length(bands),1);

for i = 1:size(bands,1)

idx = find(f >= bands(i,1) & f

bandPower(i) = trapz(f(idx), pxx(idx));

end

% Display results

for i = 1:length(bandNames)

fprintf('%s band power: %.2f\n', bandNames{i}, bandPower(i));

end

```

Time-Frequency Analysis

Wavelet transforms provide detailed time-frequency representation.

```matlab

% Using Continuous Wavelet Transform

[cwtCoeffs, frequencies] = cwt(cleanEEG, samplingRate);

figure;

imagesc(timeVector, frequencies, abs(cwtCoeffs));

axis xy;

xlabel('Time (s)');

ylabel('Frequency (Hz)');

title('Wavelet Time-Frequency Representation');

colorbar;

```

Advanced EEG Signal Analysis Techniques in MATLAB

Beyond basic filtering and feature extraction, MATLAB enables advanced analysis methods such as connectivity measures, machine learning classification, and source localization.

Connectivity Analysis

Assess the synchronization between different brain regions using coherence:

```matlab

```
% Assume signals from two channels: signal1 and signal2

[coh, f] = mscohere(signal1, signal2, windowSize, windowSize/2, [], samplingRate);

figure;

plot(f, coh);

xlabel('Frequency (Hz)');

ylabel('Coherence');

title('EEG Signal Connectivity');

...

```

## Classification for Brain-Computer Interfaces

Extract features and train classifiers:

```
```matlab

% Example feature matrix and labels

features = [bandPower1, bandPower2, ...]; % Derived from multiple channels

labels = [0, 1, 0, 1, ...]; % Example labels

% Train a classifier

classifier = fitcsvm(features, labels);

% Predict new data

predictedLabels = predict(classifier, newFeatures);

...

```

Source Localization

Using algorithms like sLORETA requires specialized toolboxes, but MATLAB can interface with

these tools to localize brain sources based on EEG data.

Visualization and Interpretation of EEG Data in MATLAB

Visualization is crucial for interpreting EEG signals. MATLAB provides multiple plotting tools to display time series, spectra, topographies, and connectivity maps.

Topographical Maps

Use EEGLAB functions or custom scripts to visualize scalp distributions:

```
```matlab

% Assuming data from multiple channels

topoplot(bandPower, channelLocations);

title('Alpha Band Power Topography');

```
```

Event-Related Potentials (ERP)

Average EEG responses to stimuli:

```
```matlab

% Segment data into epochs

epochs = eeg_epoching(rawEEG, eventMarkers, preTime, postTime, samplingRate);

% Compute average ERP

ERP = mean(epochs, 3);

plot(timeEpoch, ERP);

xlabel('Time (s)');

ylabel('Amplitude (\muV)');
```

```
title('Event-Related Potential');
```

```
```
```

Conclusion: MATLAB as an Essential Tool for EEG Signal Analysis

Using MATLAB code for EEG signal analysis offers a versatile and powerful approach to neuroscientific research and clinical applications. Its rich ecosystem of built-in functions, toolboxes, and community-developed plugins like

MATLAB Code for EEG Signal Analysis: An In-Depth Guide

Electroencephalography (EEG) signals provide a window into the human brain's electrical activity, offering invaluable insights for neuroscience, clinical diagnostics, brain-computer interfaces (BCIs), and cognitive research. MATLAB, with its robust computational environment and extensive toolbox ecosystem, has become a popular platform for EEG signal processing. This comprehensive review explores MATLAB code tailored for EEG analysis, covering everything from data acquisition and preprocessing to advanced feature extraction and visualization.

Understanding EEG Data and MATLAB's Role

EEG signals are typically recorded as time-series data across multiple channels, each representing the electrical activity of a specific scalp location. MATLAB's strength lies in its ability to handle large datasets, perform complex mathematical operations, and visualize data interactively.

Key advantages of using MATLAB for EEG analysis include:

- Built-in functions for signal processing (e.g., filtering, Fourier analysis)
- Toolboxes such as Signal Processing Toolbox and EEGLAB
- Custom scripting capabilities for tailored workflows
- Compatibility with various data formats and hardware interfaces

Data Acquisition and Importing EEG Data in MATLAB

Before processing, EEG data must be imported into MATLAB. Data formats vary depending on the acquisition system; common formats include `.mat`, `.edf`, `.bdf`, `.set` (EEGLAB format), and proprietary formats.

Importing Data Examples

- Loading MATLAB `.mat` Files:

```
``matlab

% Load pre-recorded EEG data stored in a .mat file

load('eeg_data.mat'); % assuming variables 'EEG', 'fs', 'labels' exist

% 'EEG' is a matrix (channels x samples)

% 'fs' is the sampling frequency

``
```

- Using EEGLAB to Import Data:

```
``matlab

% Start EEGLAB

eeglab;

% Import data (e.g., EDF format)

EEG = pop_biosig('subject1.edf');

% Visualize data

pop_eegplot(EEG, 1, 1, 0);

``
```

- Reading Other Formats:

- For `.bdf` or `.edf` files, functions like `pop_biosig()` or `pop_importdata()` are highly effective.

Preprocessing EEG Data in MATLAB

Preprocessing is essential to enhance signal quality, remove noise, and prepare data for analysis.

Common Preprocessing Steps

- Filtering: To remove unwanted frequency components
- Artifact Removal: Eliminating eye blinks, muscle artifacts, and line noise
- Re-referencing: To improve signal interpretability
- Segmentation: Dividing continuous data into epochs

Filtering Techniques

- Bandpass Filtering:

```
```matlab

% Design a bandpass filter (e.g., 1-40 Hz)

fs = 256; % example sampling rate

bpFilt = designfilt('bandpassiir','FilterOrder',4, ...

'HalfPowerFrequency1',1,'HalfPowerFrequency2',40, ...

'SampleRate',fs);

% Apply filter

filteredEEG = filtfilt(bpFilt, EEG);

```
```

- Notch Filtering (Line Noise Removal):

```
```matlab

% Design a notch filter at 50 Hz

d = designfilt('bandstopiir','FilterOrder',2, ...
```

```
'HalfPowerFrequency1',49,'HalfPowerFrequency2',51, ...
```

```
'SampleRate',fs);
```

```
% Apply filter
```

```
notchedEEG = filtfilt(d, filteredEEG);
```

```
...
```

- Using EEGLAB's Filtering Functions:

```
```matlab
```

```
EEG = pop_eegfiltnew(EEG,1,40); % Bandpass 1-40 Hz
```

```
EEG = pop_eegfiltnew(EEG,48,52,'revfilt',1); % Notch at 50 Hz
```

```
...
```

Artifact Removal Strategies

- Manual Inspection: Visual identification of artifacts

- Automated Algorithms: Using ICA (Independent Component Analysis) or algorithms like ASR (Artifact Subspace Removal)

ICA Example:

```
```matlab
```

```
% Run ICA to identify artifacts
```

```
EEG = pop_runica(EEG, 'extended',1);
```

```
% Visualize components
```

```
pop_eegplot(EEG, 1, 1, 0);
```

```
% Remove artifact components
```

% e.g., remove components 1 and 3

```
EEG = pop_subcomp(EEG, [1 3], 0);
```

```
...
```

## Feature Extraction from EEG Signals

Extracting meaningful features is pivotal for subsequent analysis such as classification or pattern recognition.

### Common Features and MATLAB Implementations

- Power Spectral Density (PSD):

Understanding the distribution of power across frequency bands.

```
```matlab
```

```
% Using pwelch
```

```
window = hamming(256);
```

```
noverlap = 128;
```

```
nfft = 512;
```

```
[pxx,f] = pwelch(EEG(ch, :), window, noverlap, nfft, fs);
```

```
% Plot PSD
```

```
plot(f,10log10(pxx));
```

```
xlabel('Frequency (Hz)');
```

```
ylabel('Power/Frequency (dB/Hz)');
```

```
title('PSD Estimate');
```

```
...
```

- Band Power Calculation:

Calculate power within specific bands:

```
```matlab

% Define frequency bands

delta = [1 4];

theta = [4 8];

alpha = [8 13];

beta = [13 30];

% Use bandpower function

delta_power = bandpower(EEG(ch, :), fs, delta);

theta_power = bandpower(EEG(ch, :), fs, theta);

alpha_power = bandpower(EEG(ch, :), fs, alpha);

beta_power = bandpower(EEG(ch, :), fs, beta);

```
```

- Time-Domain Features:

- Mean, variance, skewness, kurtosis
- Zero-crossing rate

```
```matlab

meanVal = mean(EEG(ch, :));

varVal = var(EEG(ch, :));
```

```
skewVal = skewness(EEG(ch, :));
```

```
kurtVal = kurtosis(EEG(ch, :));
```

```
...
```

#### - Wavelet Features:

Wavelet transforms capture both time and frequency information, suitable for transient EEG events.

```
```matlab
```

```
% Using the Wavelet Toolbox
```

```
wname = 'db4';
```

```
[c,l] = wavedec(EEG(ch, :), 4, wname);
```

```
% Extract detail coefficients
```

```
details = detcoef(c, l, 1); % first level detail
```

```
% Calculate energy
```

```
energyDetail = sum(details.^2);
```

```
...
```

Advanced EEG Signal Processing Techniques in MATLAB

Beyond basic features, advanced techniques facilitate deeper analysis.

Time-Frequency Analysis

- Short-Time Fourier Transform (STFT):

```
```matlab
```

```
% Using spectrogram
```

```
spectrogram(EEG(ch, :), window, noverlap, nfft, fs, 'yaxis');
```

```
title('Spectrogram of EEG Channel');
```

```
...
```

- Wavelet Transform: For transient features

```
```matlab
```

```
cwt(EEG(ch, :), 'amor', fs);
```

```
title('Continuous Wavelet Transform');
```

```
...
```

Connectivity and Network Analysis

- Coherence: Measures synchronization between channels

```
```matlab
```

```
[coh, f] = mscohere(EEG(ch1, :), EEG(ch2, :), window, noverlap, nfft, fs);
```

```
plot(f, coh);
```

```
xlabel('Frequency (Hz)');
```

```
ylabel('Coherence');
```

```
title('Channel Coherence');
```

```
...
```

- Graph Metrics: Using connectivity matrices to analyze brain networks

## Visualization of EEG Data in MATLAB

Visualization aids in interpreting EEG signals and verifying processing steps.

- Raw and Processed Signals:

```
```matlab

time = (0:length(EEG)-1)/fs;

figure;

plot(time, EEG(ch, :));

xlabel('Time (s)');

ylabel('Amplitude (\muV)');

title('EEG Signal');

```
```

- Topographical Maps:

Using EEGLAB's `pop_topoplot()`:

```
```matlab

pop_topoplot(EEG, 0, [start_time end_time]fs, 'EEG Topography', 0, 1);

```
```

- Spectrograms and Time-Frequency Representations:

```
```matlab

spectrogram(EEG(ch, :), window, noverlap, nfft, fs, 'yaxis');

title('EEG Spectrogram');
```

...

Automation and Custom Pipelines in MATLAB

Building reproducible workflows often involves scripting and modular functions.

Example Workflow:

- Import raw data
- Filter data
- Remove artifacts
- Segment into epochs
- Extract features
- Save features for classification

Sample Skeleton Code:

```
```matlab

% Step 1: Import

EEG = importEEGData('subject.edf');

% Step 2: Filter

EEG_filtered = bandpassFilter(EEG, fs, [1 40]);

% Step 3: Artifact removal via ICA

EEG_clean = removeArtifactsICA(EEG_filtered);

% Step 4
```

**Question** How can I preprocess EEG signals using MATLAB? You can preprocess EEG signals in MATLAB by importing the data, applying filters (like bandpass or notch filters), removing artifacts, and normalizing the signals. MATLAB's Signal Processing Toolbox offers functions such as 'filter', 'bandpass', and 'notch' to facilitate this process. What are common MATLAB toolboxes used for EEG signal analysis? Common MATLAB toolboxes for EEG analysis include the Signal Processing Toolbox for filtering and feature extraction, the Brain Connectivity Toolbox for connectivity analysis, and the EEGLAB toolbox (available as an open-source plugin) for

comprehensive EEG data processing and visualization. How do I implement an EEG signal classification algorithm in MATLAB? Begin by extracting features from your EEG data, such as power spectral density or wavelet coefficients. Then, use MATLAB's machine learning functions like 'fitcdiscr', 'fitcsvm', or 'trainNetwork' to train classifiers such as SVM or neural networks. Validate your model with cross-validation to ensure accuracy. Can MATLAB be used to visualize EEG signals effectively? Yes, MATLAB provides plotting functions like 'plot', 'spectrogram', and 'topoplot' (via EEGLAB) to visualize EEG signals, their spectral content, and scalp distributions, enabling comprehensive analysis and interpretation of EEG data. What is the best way to load and save EEG data in MATLAB? EEG data can be loaded into MATLAB using functions like 'load' for MAT files, or custom scripts for formats like EDF or CSV. To save processed data, use 'save' to store variables in MAT files or export to formats like CSV for further analysis or sharing.

**Related keywords:** EEG signal processing, MATLAB EEG analysis, EEG signal filtering, MATLAB script for EEG, EEG data visualization, MATLAB EEG toolbox, EEG feature extraction MATLAB, MATLAB EEG signal preprocessing, EEG signal analysis MATLAB code, MATLAB brain wave analysis