Journal of Clinical Monitoring and Computing (2022) 36:609–621
https://doi.org/10.1007/s10877-021-00771-4

REVIEW PAPER

Power spectrum and spectrogram of EEG analysis during general anesthesia: *Python*-based computer programming analysis

Teiji Sawa[1] · **Tomomi Yamada**[1] · **Yurie Obata**[2]

Received: 26 March 2021 / Accepted: 13 October 2021 / Published online: 29 October 2021

Abstract

The commonly used principle for measuring the depth of anesthesia involves changes in the frequency components of the electroencephalogram (EEG) under general anesthesia. Therefore, it is essential to construct an effective spectrum and spectrogram to analyze the relationship between the depth of anesthesia and the EEG frequency during general anesthesia. This paper reviews the computer programming techniques for analyzing the spectrum and spectrogram derived from a single-channel EEG recorded during general anesthesia. A periodogram is obtained by repeating a Fourier transform on EEG segments separated by short time intervals, but spectral leakage (i.e., dissociation from the true spectrum) occurs as a consequence of unnatural segmentation and noise. While offsetting the securing of the dynamic range, practical analyses of the adaptation of the window function are explained. Finally, the multitaper method, which can suppress artifacts caused by the edges of the analysis segments, suppress noise, and probabilistically infer values that are close to the real power spectral density, is explained using practical examples of the analysis. All analyses were performed and all graphs plotted using *Python* under *Jupyter Notebook*. The analyses demonstrated the effectiveness of *Python*-based programming under the integrated development environment *Jupyter Notebook* for constructing an effective spectrum and spectrogram for analyzing the relationship between the depth of anesthesia and EEG frequency analysis in general anesthesia.

Keywords EEG · General anesthesia · Spectrum analysis · Spectrogram · Multitaper method

1 Introduction

Measuring the electroencephalogram (EEG) of patients during anesthesia management has become popular in the clinical setting for monitoring the depth of general anesthesia. Several methods of EEG acquisition and processing have been developed and approved for clinical use. Among them, commercially available depth-of-anesthesia monitors, such as the Bispectral index® (BIS) monitor (Medtronic, Minneapolis, USA) and Sedline ® (Masimo, Irvine, USA), which analyze oscillatory behavior of the frontal EEG derived from sensors placed on the patients' forehead, are widely used around the world [1–3]. Measuring anesthetic depth using these monitors involves changes in the frequency components of the frontal EEG under general anesthesia. Most of the spectral power is concentrated above 10 Hz in the high alpha, beta, and gamma bands in the awake state. However, under general anesthesia using propofol or sevoflurane, the low-frequency components of the signal increase in amplitude, and the spindle wave (around 10 Hz), theta wave, and delta wave components emerge.

In a recent experiment, part of the reason for the hypersynchronous alpha rhythm observed in the frontal cortex during general anesthesia was explained: the action of anesthetics on GABA receptors in the nucleus reticularis prompts further discussion of the thalamus and its connectivity to the cortex in anesthetic-induced unconsciousness [4–7]. However, the exact mechanism of this phenomenon is still uncertain. Although there are some similarities in the spectral changes between anesthesia and sleep, the characterization of anesthesia-specific EEG changes that identify a lack of awareness and prevent excessively profound state is essential for progress in research to measure the depth of anesthesia.

The analysis of the EEG obtained from a single channel is a typical univariate time series analysis. An EEG

✉ Teiji Sawa
anesth@koto.kpu-m.ac.jp

[1] Department of Anesthesiology, Kyoto Prefectural University of Medicine, Kyoto, Japan

[2] Department of Anesthesia, Yodogawa Christian Hospital, Osaka, Japan

is a time-axis record of minute voltage changes measured through electrodes. In clinical EEG diagnostics, voltage changes derived from a wide range of brain surfaces are acquired using multiple channels, and spatial analysis is added. Generally, there are two types of analysis techniques for extracting EEG characteristics as time-series data: (1) the time-domain technique and (2) the frequency-domain technique. The focus of this review is the frequency-domain technique applied to single-channel frontal EEG derived from a sensor attached to the patients' forehead, and the computer analysis of EEG changes during general anesthesia is described.

With the recent advancement of the central processing unit of personal computers and software technology, the complex analysis environment that requires high-speed arithmetic processing has changed drastically. Previously, it was necessary to program complex mathematical algorithms using computer languages, such as Fortran, C, and Java, all of which require a great deal of complicated syntactic knowledge, and compiling and executing techniques. However, in recent years, more user-friendly high-level interactive languages, such as *Python* and the *R* language, have become popular. Using their integrated development environments, such as *Jupyter Notebook* and *Apache Zeppelin*, programming and graphing the mathematical procedures for analyses has become as easy as writing and testing the program code using a pencil and eraser in a notebook. By contrast, particularly for clinician-researchers who are not familiar with software engineering, it is still a significant hurdle to work with computer languages and data processing, particularly at the start of a study. The theoretical bases of many EEG analysis techniques provided in this review have already been established. The purpose of this review is to provide *Python* algorithms for mathematical calculations and graphing techniques in the *Jupyter Notebook* environment for spectrums and spectrograms, with practical examples to enhance research concerning the depth of anesthesia and EEG. All analyses were performed and all graphs plotted using *Python* (version 3.5) under *Jupyter Notebook* in the *Anaconda* environment (Table 1). The EEG Analyzer (version 54_GP), developed by the author, was used to acquire the EEG from the BIS monitor [8, 9].

2 Theoretical background

2.1 Digitized EEG

The voltage change measured through the EEG electrode is acquired as a continuous analog signal but is digitized by an analog-to-digital converter. The discrete Fourier transform (DFT) and fast Fourier transform (FFT, an algorithm for computing the DFT), which are the practical application of Fourier transforms in digital signal processing of the time-sampled waveform, are used to calculate the frequency component [10, 11]. In the EEG measurement of clinical diagnostics, digitization is typically performed at a high sampling frequency (typically 1000–5000 Hz). By contrast,

Table 1 *Python* operating environment and *Python* libraries useful for EEG frequency analysis

Library	Usage	URL
Python	*Python* Software Foundation	https://www.python.org
Anaconda	a platform for providing an environment for data science, including *Python*	https://www.anaconda.com
Jupyter Notebook	Data analysis environment for programming by *Project Jupyter*	https://jupyter.org
pandas	*Python* data analysis library. Data structures and operations for manipulating mathematical tables and time-series data	https://pandas.pydata.org
matplotlib	Graph drawing	https://pandas.pydata.org
numpy	Numerical calculation	https://matplotlib.org
SciPy	Scientific and engineering calculation	https://www.scipy.org
nitime	Time series data analysis in neuroscience	http://nipy.org/nitime/
mtspec	Multitaper spectral estimation	https://krischer.github.io/mtspec/
libtfr	Multitaper time frequency reallocation (TFR) spectrogram (Installation of C++FFT calculation library *FFTW* http://www.fftw.org is required)	https://pypi.org/project/libtfr/ https://melizalab.github.io/libtfr/
librosa	Music and audio analysis	https://librosa.org
statsmodels	*Python* statistical model module	https://www.statsmodels.org/stable/index.html

Anaconda and *Jupyter Notebook* are *Python* execution environments. *pandas*, *matplotlib*, *numpy* and *SciPy* are almost standard libraries used in data science. For frequency analysis, the window function *signal()* and the FFT function *fft()* in *fftpack* are from the *scipy* library, and the *tfr_spec()* function from the *libtfr* library can be used. For the multitaper method, the *dpss()* function from the *mtspec* library, the *mfft_dpss()* function from the *libtfr* library, the *spectrogram()* function from *matplotlib*, or the *display.specshow()* function from the *librosa* library can be used to create the spectrogram. The *statsmodels* library and *nitime* library are useful for analysis using autoregressive models the sampling frequency 128 Hz is sufficient for analyzing the frequency components of EEG waves because, in the DFT, the Nyquist frequency (Eq. 1), which is the upper limit of the maximum frequency component, is half the sampling frequency of the Nyquist–Shannon sampling theorem.

$$Nyquist\, frequency\, f_n = \frac{1}{2} \bullet sampling\, frequency\, f_s. \tag{1}$$

The size of the analysis segment determines the width of the frequency bin because the size of the frequency bin in DFT is the sampling frequency divided by the data size of the analysis segment:

$$Frequency\, bin = f_s/N_{record} = 1/(N_{record} \times \Delta t_{sample}) \tag{2}$$

where f_s :$samplingfrequency$,N_{record} : $numberofdatapoints$, and Δt_{sample} : sampling interval time.

Therefore, increasing the size of the analysis segment improves the estimated frequency resolution in the power spectrum but sacrifices time responsiveness. When the sampling frequency is 128 Hz, the analysis segment containing 128 data points (1 s) (Fig. 1a, b) creates a 1.0 Hz frequency bin (128 Hz/128 data points = 1.0 Hz/bin) between 0 and 64 Hz (= the Nyquist frequency of 128 Hz sampling). Conversely, to create a 0.25 Hz frequency bin at a 128 Hz sampling frequency, the data size should be 512 data points (= 4 s), which is the condition used in the examples in this review.

In response to an EEG data request through the serial input/output port of the BIS monitor, data packets (for each channel) are transmitted at the rate of eight packets per sec, where each packet contains 16 samples (i.e., 128 samples per sec). Each sample is a 32-bit signed integer representing a digitized μV value, in the range − 32,768 to 32,767. The waveform is reconstructed by the above set of signed integers using offset correction in the processing by the receiver. Note that when analyzing EEG data, it is necessary to pay attention to the data format. For example, the European Data Format (EDF) and European Data Format 'plus' (EDF +) are popular binary file formats for EEG data adopted by many EEG measuring devices. For reference, the *Python* code used to convert the EEG data saved in the EDF file to microvolt data is shown in the supplementary data SI-1 (EDF2rawEEG.pdf) [12, 13].

In the acquisition of EEG data during actual surgery, noise signals from the electric knife used by the surgeon may be mixed. When the patient is awake, the signals from the electromyogram derived from the orbicularis oculi muscle may also be mixed in the EEG signals. Before the application of spectral analysis to the EEG data, preprocessing using an algorithm that excludes outliers and performs spline complementation (this is not explained in this paper because of space limitations) is typically required. The supplementary *Python* code shown in spectral_analysis_2_deep_anesth.pdf contains the preprocessing steps for the elimination of outliers and spline complementation. The recorded EEG might be artificially affected by other manipulations of the monitor device and form signal distortion (so-called clipping), as recently reported by the SedLine® monitor, which may strongly affect the retrospective analysis of the EEG signal [14].

2.2 Discrete Fourier transform

The mathematical technique used for frequency analysis is the Fourier transform [10]. This transformation represents any function *f*(*x*) as a linear combination of trigonometric functions with different frequencies and amplitudes:

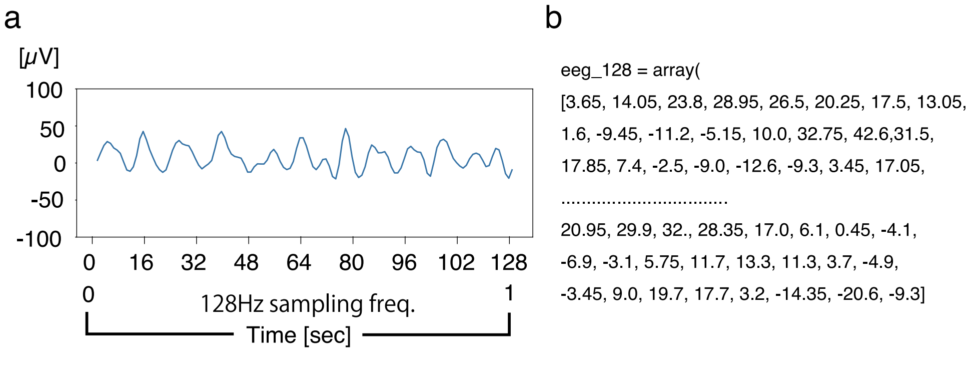
$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} \left(a_n \cos\frac{2\pi nx}{T} + b_n sin\frac{2\pi nx}{T} \right), \tag{3}$$

where the Fourier coefficients (a_0, a_n, b_n in the interval [*0*, *T*]) are defined as follows:

$$a_0 = \int_0^T f(x)dx, a_n = \frac{2}{T}\int_0^T f(x)cos\frac{2\pi nx}{T}dx,$$

$$b_n = \frac{2}{T}\int_0^T f(x)sin\frac{2\pi nx}{T}dx.$$

The Fourier transform formula can also be expressed by the following complex exponential function, according to Euler's formula $e^{i\theta} = cos\theta + i \cdot sin\theta$:



```
eeg_128 = array(
[3.65, 14.05, 23.8, 28.95, 26.5, 20.25, 17.5, 13.05,
1.6, -9.45, -11.2, -5.15, 10.0, 32.75, 42.6,31.5,
17.85, 7.4, -2.5, -9.0, -12.6, -9.3, 3.45, 17.05,
................................
20.95, 29.9, 32., 28.35, 17.0, 6.1, 0.45, -4.1,
-6.9, -3.1, 5.75, 11.7, 13.3, 11.3, 3.7, -4.9,
-3.45, 9.0, 19.7, 17.7, 3.2, -14.35, -20.6, -9.3]
```

Fig. 1 Digitized EEG and analysis window. **a** EEG waveform restored from digital data (1-s duration, sampling frequency 128 Hz, 23-year-old male, general anesthesia by inhalation of sevoflurane, BIS value = 55). **b** Digitized EEG voltage values (μV)

$$f(x) = \sum_{n=-\infty}^{\infty} C_n \exp\left(\frac{2\pi inx}{T}\right), \tag{4}$$

where $C_n = \frac{1}{T}\int_0^T f(x)\exp\left(-\frac{2\pi inx}{T}\right)dx$.

The frequency power distribution called the power spectrum, whose density is the power spectral density (PSD), is estimated by the DFT, which commonly uses the FFT algorithm [11]. A simple estimate of the PSD using the DFT is called a periodogram, which is obtained by repeating the Fourier transform on EEG segments separated by short time intervals.

2.3 Processing using a window function

In the Fourier transform, both the function $f(x)$ and the trigonometric function are defined in infinite intervals $(-\infty, \infty)$ [10]. However, when the actual EEG data are Fourier transformed, an EEG of infinite length cannot be handled; therefore, data in the finite interval $[T_1, T_2]$ (comprising 2^n signals) are processed, and those outside the interval are ignored. When the input waveform is time-sampled, instead of continuous, the analysis is usually done by applying a window function and then a discrete Fourier transform (DFT). This is equivalent to considering the function $f(x)$ to be zero outside the interval. The product of the function $f(x)$ and the function

$$w(x) = \{1(\text{if}: T_1 \le x \le T_2), 0(\text{if}: x < T_1, \text{ or } T_2 < x)\} \tag{5}$$

is then obtained, and the Fourier transform is applied to the product (Fig. 2a).

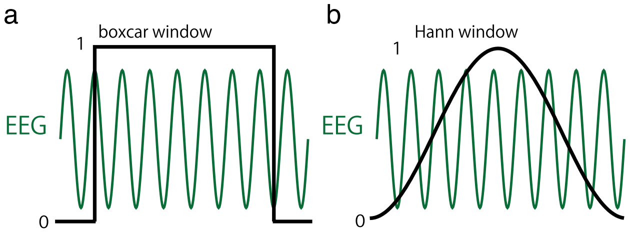


Fig. 2 EEG and analysis window. **a** Boxcar window. **b** Hann window

Even if the signal is continuous, depending on the segment selected for the Fourier transform, the data at the edge undergo a sudden unnatural cut-processing. A Fourier transform applied to the interval $[T_1, T_2]$ is equivalent to fitting a rectangular (boxcar) window (Fig. 3a), and the analysis result includes the main center frequency (main lobe in Fig. 3g), and peripheral frequency bands that gradually decrease on both wings (side lobes in Fig. 3g). Multiplying the EEG signal with a boxcar window function is equivalent to convolving it in the frequency domain with the spectrum of the boxcar window, thereby producing an artificial spectrum that does not exist in the spectrum of the original wave. These artificial spectra observed at any given frequency bin are called spectral leakage. Therefore, two factors, that is, the unnatural cutting edge of the analysis window and the side lobes of the window spectrum, artificially modify the true spectrum of the original wave. The effect of the window functions on spectral leakage is discussed in Sect. 3.

Increasing the number of EEG signal samples improves the estimated frequency resolution, and increasing the sampling frequency increases the estimated maximum frequency

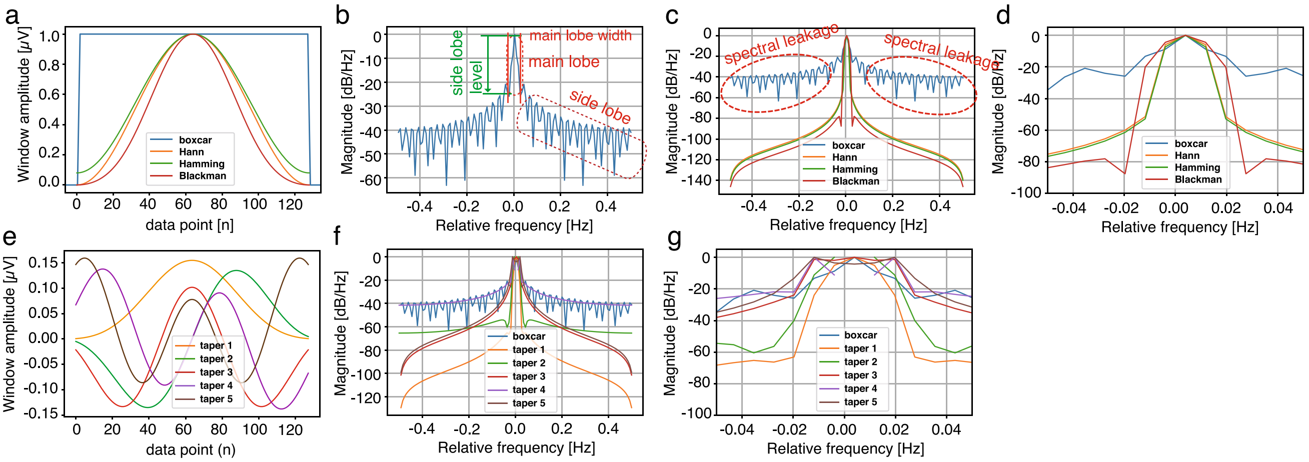


Fig. 3 Power spectrum analysis using the window function and multitaper methods. **a** Window function. Hann window: $w(x) = 0.5–0.5\cos2\pi x$, if $0 \le x \le 1$. Hamming window: $w(x) = 0.54–0.46\cos2\pi x$, if $0 \le x \le 1$. Blackman window: $w(x) = 0.42–0.5\cos2\pi x + 0.08\cos4\pi x$, if $0 \le x \le 1$. **b** Fourier analysis of the window function. **c** Enlarged view around the center frequency in the Fourier analysis of the window function itself. **d** Discrete prolate spheroid sequence (DPSS, Slepian sequence). **e.** Fourier analysis of DPSS. **f** Enlarged view around the center frequency in the Fourier analysis of DPSS. **g** Frequency characteristics of the window functions

(the Nyquist frequency). However, such actions conflict with our desire to capture and analyze changes in real time using efficient calculations. The problem caused by the unnatural edges of the analysis segment can be alleviated by applying a window that brings the function at both ends of the sample interval close to zero (Fig. 2b shows a Hann window, as an example). The function $w(x)$, to be multiplied by $f(x)$, is called a window function. Various window functions that bear the developer's name, such as the Hann window (Fig. 2b), Hamming window, and Blackman window, have been used (as shown in Fig. 3a; the Hann window is also called the Hanning window as a pun on the Hamming window):

$$\text{Hann window: } w(x) = 0.5 - 0.5cos2\pi x, if0 \leq x \leq 1 \tag{6}$$

$$\text{Hamming window: } w(x) = 0.54 - 0.46cos2\pi x, if0 \leq x \leq 1 \tag{7}$$

$$\text{Blackman window: } w(x) = 0.42 - 0.5cos2\pi x + 0.08cos4\pi x, \\ if0 \leq x \leq 1. \tag{8}$$

The Hamming window, which has better frequency resolution than the Hann window, has a narrow dynamic range and is discontinuous at both ends of the interval. The Blackman window has a poorer frequency resolution than the Hann and Hamming windows, but has a wider dynamic range. Processing with a window function that approaches zero at both ends is sometimes expressed as tapering. Figure 3a and b show tapering using the Hann, Hamming, and Blackman windows (shown in Fig. 3a by the orange, green, and red lines, respectively) and the frequency component (Fig. 3b, c) using the Fourier transform for a rectangular window without tapering (shown by the blue line in Fig. 3a). This demonstrates that tapering suppresses the peripheral frequencies of both wings at the center frequency. As another characteristic, the center peak of the window function provides the value 1.0, which means that the signal amplitudes of the data points around the central region of the analysis segment are retained, whereas those near the edge of the segment are reduced. This is both an advantage and disadvantage in the construction of the spectrum: spectral leakage from the segmental edge is reduced, whereas spectral information close to the edge tends to be ignored. Two characteristics are required for a practical window function: (1) appropriate frequency resolution (narrow main lobe) and (2) wide dynamic range (low side lobe) (Fig. 3c). A window function with a taper function achieves a wide dynamic range because of its low side lobe (Fig. 3b). There is a trade-off between these two requirements; hence, it is difficult to achieve both. Therefore, the introduction of the multitaper (MT) method, which is explained in the next section, is a solution to this dilemma.

2.4 Multitaper method

In PSD estimation using a window function, as described above, there is the problem that information near the boundary of the analysis frame is lost as a result of tapering, even with window processing. For the efficient analysis of time-dependent EEG changes, it is necessary to examine each short analysis frame and process it simultaneously to obtain the spectral information instantaneously, as far as possible. An approach is the Welch method, which uses a device to average the spectrum by applying a sliding window to various parts of the signal. Another approach is the MT method, which uses multiple taper functions, expressed by the following [15, 16]:

$$\widehat{S}^{(MT)}(f) \overset{\text{def}}{=} \frac{1}{K}\sum_{k=0}^{K-1}\widehat{S}_k^{(MT)}(f) with \widehat{S}_k^{(MT)}(f) \overset{\text{def}}{=} \Delta_t \left|\sum_{t=0}^{N-1} h_{k,t}X_t e^{-i2\pi ft\Delta_t}\right|^2, \tag{9}$$

where $h_{k,\ t}$ is the data taper of the kth spectrum estimate $\widehat{S}_k^{(MT)}(\bullet)$ and Δ_t is the sampling interval $X_t : X_0, X_1, ..., X_{N-1}$ of the stationary process $\{X_t\}$.

A comprehensive review of the power spectral analysis using the MT method was published by Babadi B and Brown EN in 2014 [16]. In the MT method, the spectrum obtained by processing data with multiple window functions (called taper functions) is weighted and averaged. The influence of spectral leakage, caused by edges and noise, is suppressed to reduce variation. The signal data near the boundary are impaired by the first taper, but this is compensated for by the subsequent tapers to give the entire spectrum an accurate relative signal level [16, 17]. Using a discrete prolate spheroid sequence (also called the DPSS or Slepian sequence) (Fig. 3d, e) as a set of tapered windows, orthogonal to each taper function, solves the problem of distorted window correction with a single window function [18].

3 Frequency analysis of simple sine and cosine waves

By performing the FFT analysis of sine and cosine waves with a fixed period, the trade-off between the decreased resolution of the band near the target frequency (narrow-band leakage) and attenuation of spectral leakage from the far band (broadband leakage) can be observed and understood. In this paper, 2-s sine waves (Fig. 4a) and cosine waves (Fig. 4b), with a period of 5 Hz and amplitude of 1 at a sampling frequency of 128 Hz, are applied to the FFT. For both the sine and cosine waves, the waveform frequency is 5 Hz. The analysis segment of the sine wave starts and ends at 0, whereas that of the cosine wave starts and ends at 1. Therefore, these waveforms are affected differently by the different edge patterns caused by their phase difference. By performing a simple FFT on both of these waveforms, the PSDs with the main component of 5 Hz are plotted in the range of 0 to 64 Hz (Fig. 4c).

When the PSDs are plotted with μV^2/Hz as the magnitude unit, the PSDs of both waveforms overlap, and the difference is too small to be recognized. However, if the PSDs are plotted in decibel/Hz (dB/Hz) units and enlarged around the 5 Hz frequency band region, the PSD difference between the sine and cosine waves becomes obvious. Although both PSDs have a sharp triangular 5 Hz peak (Fig. 4d), the PSD from the sine wave displays a wide dynamic range in the low-frequency

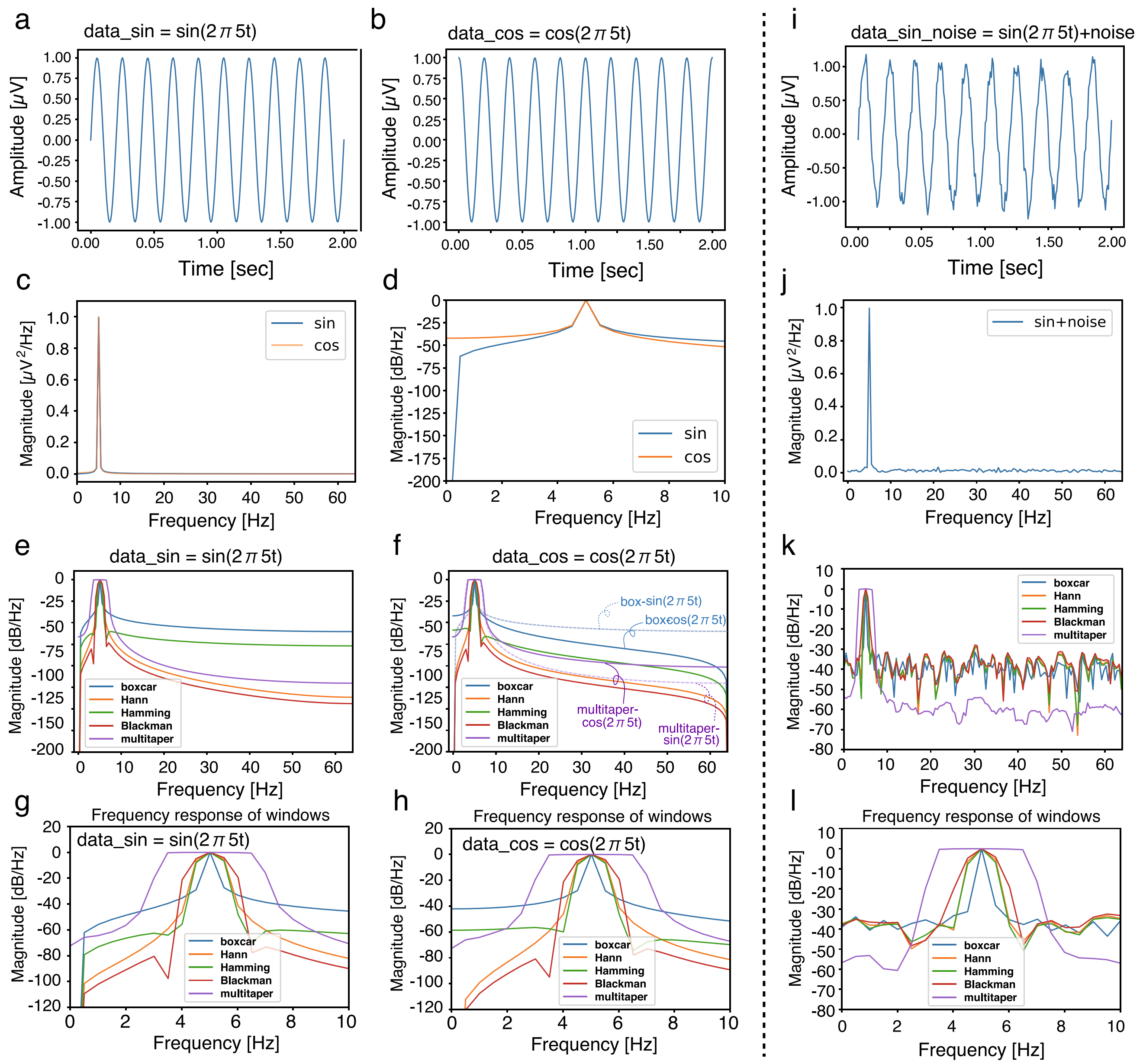


Fig. 4 Power spectrum analysis of the sine and cosine waves. Both the sine and cosine waves are 5 Hz, sampling frequency is 128 Hz, and duration is 2 s. **a** Sine wave. **b** Cosine wave. **c** Periodogram from the sine and cosine waves. **d** Periodogram from the sine and cosine waves (enlarged view around 5 Hz, in dB/Hz units). **e** Power spectrum analysis of the sine wave using the window function method and multitaper method. **f** Power spectrum analysis of the cosine wave using the window function method and multitaper method. **g** Power spectrum analysis of the sine wave using the window function method and multitaper method (enlarged view around 5 Hz, in dB/Hz units). **h** Power spectrum analysis of the cosine wave using the window function method and multitaper method (enlarged view around 5 Hz, in dB/Hz units). **i** Sine wave + noise (mean 0, standard deviation 0.1). **j** Power spectrum analysis of sine wave + noise using the periodogram. **k** Power spectrum analysis of the sine wave + noise using the multitaper method (in dB/Hz units). **l** Power spectrum analysis of the sine wave + noise using the window function method and multitaper method (enlarged view around 5 Hz, in dB/Hz units)

region, whereas the PSD from the cosine wave shows a narrower dynamic range in high-frequency areas. Figure 4e and f show a comparison of the solid blue cosine line and dotted blue sine line. This difference in PSD between the sine and cosine waves demonstrates that, when frequency analysis is performed by dividing the waveform into finite times, the phase difference affects the PSD results.

Next, either a window function (Hann, Hamming, or Blackman, Fig. 3a) or the MT method is applied to the PSD analyses of both the sine and cosine waves (Fig. 4e–l). From the application of any window function, wideband spectral leakage seen in the boxcar-treated frequency domain outside 5 Hz is reduced by both the MT method and window function processing (from the comparison between a blue boxcar-treated line and other window-treated colored lines in Fig. 4e, f). When the 0–10 Hz frequency region is enlarged, using dB/Hz units (Fig. 4g, h), the spectrum peaks are plotted as flat plateaus with a width of about 5 ± 2 Hz, particularly in the MT method. The MT method allows a decrease in frequency resolution while moderately suppressing wideband leakage. Furthermore, by tapering the edge of the analysis segment, using the window function, the difference between the sine and cosine waves is minimized in the PSD using dB/Hz units.

Finally, the effect of noise on the frequency analysis of a sine waveform is evaluated. In this analysis, white noise whose amplitude has a mean value of zero and a standard deviation of 0.1 is added to a 2-s sine wave with a period of 5 Hz and amplitude of 1 (Fig. 4i). Because of the influence of noise on the PSD, small fluctuations in amplitude, with a jagged pattern, are observed on the baseline of both wings at the peak of 5 Hz (Fig. 4j). The PSDs obtained by applying various window functions and the MT method are displayed in dB/Hz in Fig. 4k. The effects of noise on the frequency bands that constitute the baseline are observed. In the enlarged range centered at 5 Hz (Fig. 4l), the influence of this noise is large in the low-power region, even when the window functions are used. By contrast, when the MT method is used, a high dynamic range is maintained across a wide range of both peak wings, although the resolution of the peak at 5 Hz is inferior to that obtained when other windows are used (Fig. 4k, l).

4 Spectrum estimation example using an autoregressive process

To understand spectral leakage caused by FFT analysis and the influence of the MT method, time-series data with known spectral characteristics, generated from the autoregressive process, are analyzed. Using the autoregressive integrated moving average model (ARIMA) from the *Python* statistical library *statsmodels* (Table 1), a waveform with a known frequency response is generated. The effect of the FFT on the PSD of the EEG is then investigated.

Using the ARIMA formula,

$$y_t = 1 + -2.76 \times y_{-1} + 3.8 \times y_{-2} + -2.65 \times y_{-3} + 0.92 \times y_{-4}, \quad (10)$$

a waveform consisting of 512 data points is generated by the *generate_sample()* function of *statsmodels* (Fig. 5a). This waveform is regarded as a digitized EEG at 128 Hz with a duration of 4 s. Using the *freq_response()* function of the *Nitime* library algorithms.spectral (Table 1), precise spectral analysis is performed under the assumption that the signal is from the infinite impulse response for the time-series sequence of this autoregressive process (as shown by the solid blue line in Fig. 5b). The PSD of the periodogram is then obtained by the *periodogram()* function. The PSD (shown by the solid orange line in Fig. 5b) along the true spectrum is obtained at 40 Hz or lower. However, in the region above 40 Hz, the line of the calculated PSD is slightly dissociated from the line of the true PSD. After the Hann window is applied, the dissociation is suppressed, which suggests that it was caused by the boxcar window (Fig. 5c).

Finally, the MT method of the generated sequence from this ARIMA model is performed by *Nitime*'s algorithms. spectral library using its *multi_taper_psd()* function. The PSD calculated by the MT method is more similar to the true PSD than the PSD derived from Hann window processing (Fig. 5d). The MT method using the *multi_taper_psd()* function enables the calculation of 95% confidence in the spectrum using either the $2 \times K_{\max}$ degrees of freedom of the chi-square model (Fig. 5e) or the iterative adaptation method and the jackknife technique for spectral bias correction (Fig. 5f) (details of *Nitime* are presented in Table 1). In this respect, the advantage of the MT method is that it can suppress spectral leakage, retain the signal power near the edges, suppress noise, and probabilistically infer values that are close to the real PSD.

5 EEG frequency analysis in practice

In this section, the above FFT analysis is applied to an EEG acquired from a patient through a depth-of-anesthesia monitor during general anesthesia (23-year-old man, urachal remnant removal, sevoflurane inhalation general anesthesia, BIS value = 55). Because the EEG was composed of 128-Hz digitized data, the 128-s EEG data contained 16,384 data points at a sampling frequency of 128 Hz (Fig. 6a). For this 128-s EEG data, the FFT algorithm was repeatedly applied to the short data segments with a certain percentage overlap with the next segment along the time course. The 4-s data used in this study comprised 512 data points (Fig. 6b). Periodogram analysis was performed using the FFT function *fft()*

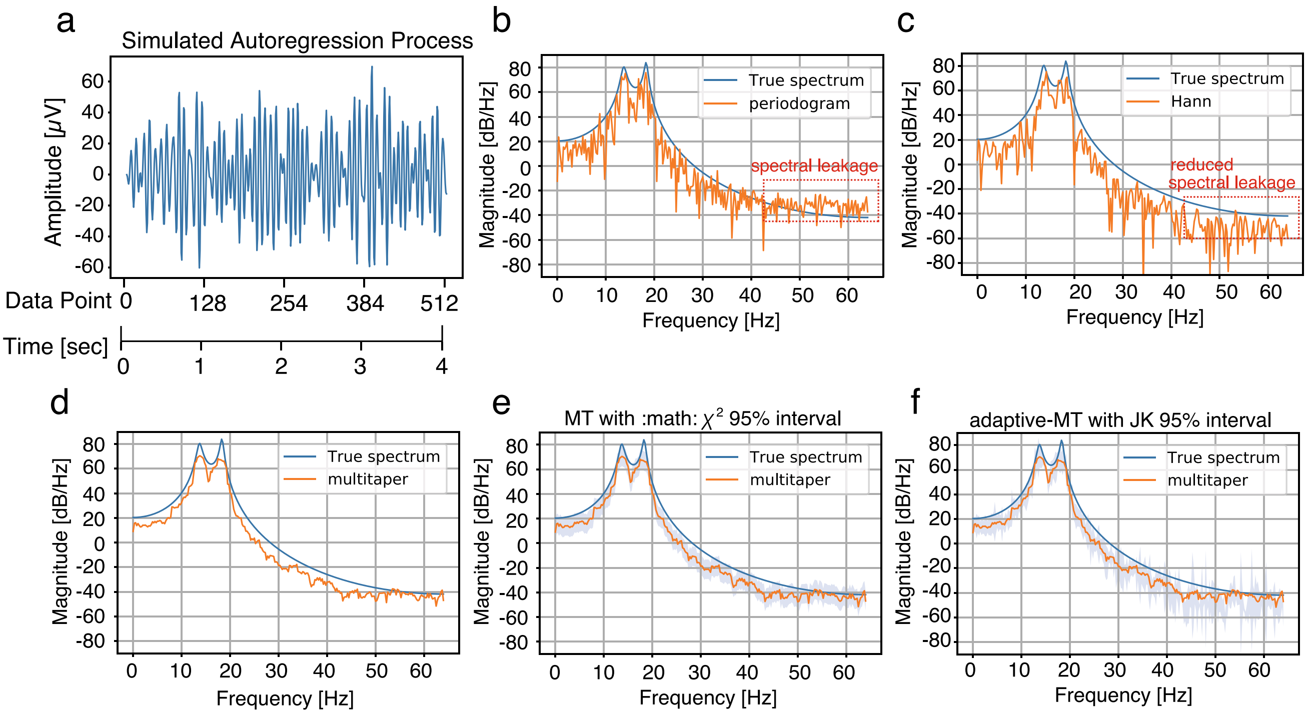


Fig. 5 Power spectrum analysis for generated waveforms using the autoregressive integrated moving average model (ARIMA). **a** Waveform from the ARIMA model (512 data points, equivalent to 4 s with a sampling frequency of 128 Hz). **b** Power spectral density (PSD) analysis for the time-series sequence of the regression process. True power spectrum (solid blue line), periodogram (solid orange line), and spectral leakage (dotted red line box). **c** PSD using the adaptation of the Hann window function. **d** PSD using the adaptation of the multitaper method. **e** Added 95% confidence interval estimation from the $2 \times K_{\max}$ degrees of freedom of the chi-square model to the PSD using the multitaper method. **f** Estimated 95% confidence interval of spectrum using either iterative adaptation or jackknifing to PSD using the multitaper method

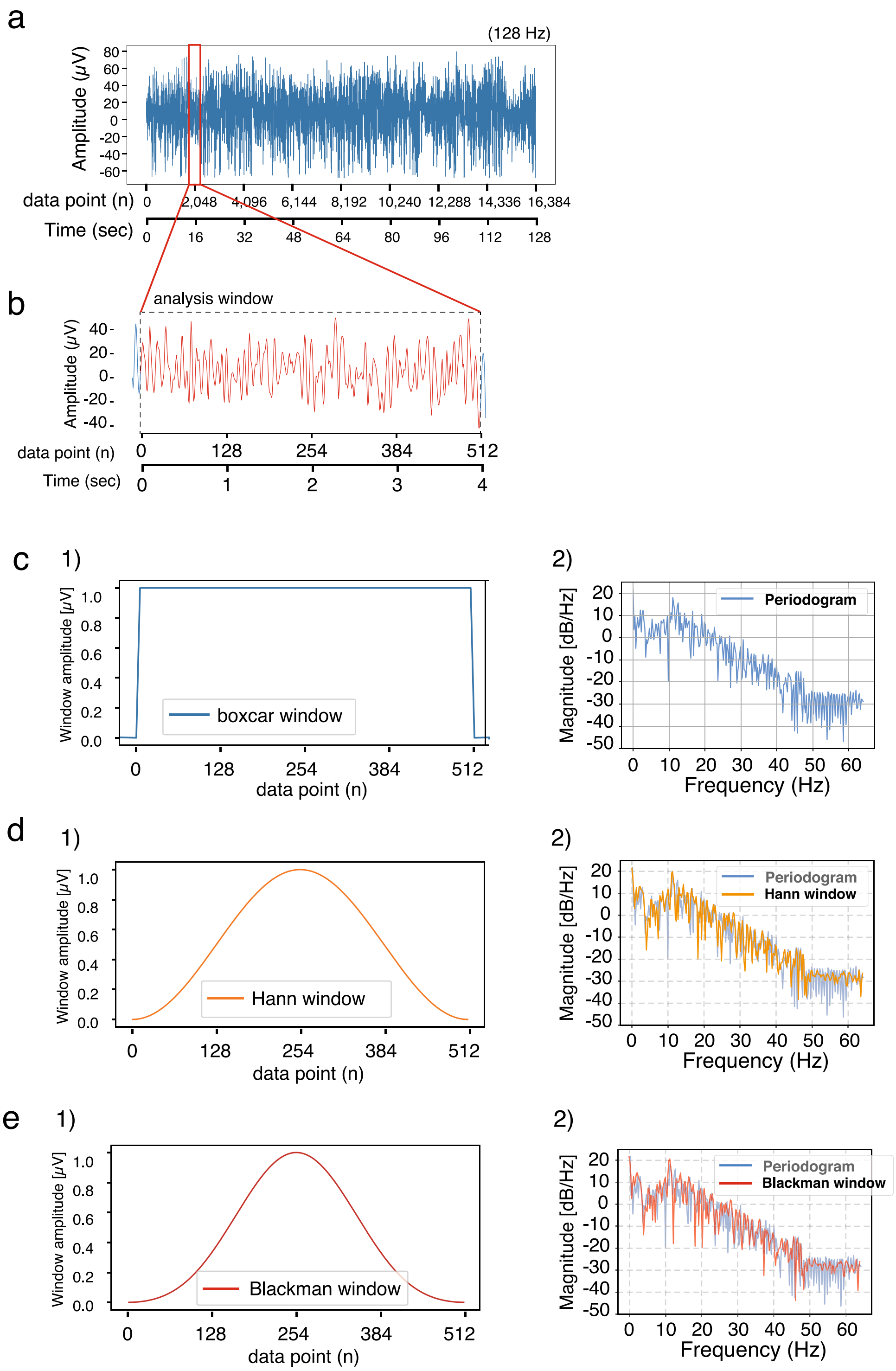
from the FFTpack package of the *Python* scientific calculation library *SciPy*. The analysis without any window process was equal to applying a boxcar window (Fig. 6c-1). The FFT captured the characteristic pattern of the sevoflurane anesthesia, with the emergence of spindles in the alpha-wave region near 10–12 Hz. In the high-frequency band, that is, 45 Hz or higher, a large fluctuation in low power magnitude was visible (Fig. 6c-2). The Hann or Blackman window was then applied using the *SciPy* function *signal* (Fig. 6d-1, e-1). Both windows started from zero and, after the center peaks (which had a value of 1.0), ended up at zero. Multiplying these window functions with the EEG segmental data helped to reduce the fluctuation seen in the high-frequency band of the boxcar-treated frequency domain (Fig. 6d-2, e-2).

Finally, the MT method was applied to this EEG analysis. The EEG analysis segment (Fig. 7a upper) was multiplied by five Slepian sequences [18] (Fig. 7a lower, b, and c) as tapers, FFT was performed on each taper process (Fig. 7d), and the average of the FFT results was calculated (Fig. 7e). The vertical vibration of the power of each frequency, recognized in all frequency bands, was alleviated, and a continuous and smooth PSD was obtained; that is, the MT method was able to eliminate all the noise and minimize the influence of unnatural edge processing on the analysis segment.

6 Configuration of the spectrogram

To evaluate how the frequency changes over time in EEG during general anesthesia (23-year-old man, urachal remnant removal, sevoflurane inhalation general anesthesia), a spectrogram that visualizes the PSD differences in chronological order needs to be configured. The spectrogram is a

Fig. 6 Power spectral analysis using the adaptation of the window functions.**a** EEG wave (23-year-old male, under general anesthesia with sevoflurane inhalation, BIS = 55). Sampling frequency 128 Hz, 512 data points for 4 s. **b** Analysis segment of the above EEG wave. Sampling frequency 128 Hz, 512 data points for 4 s. **c** (1) Boxcar window (rectangular window). (2) PSD from the periodogram. **d** (1) Hann window. (2) PSD after the application of the Hann window (orange line). **e** (1) Blackman window. (2) PSD after the application of the Blackman window (red line)



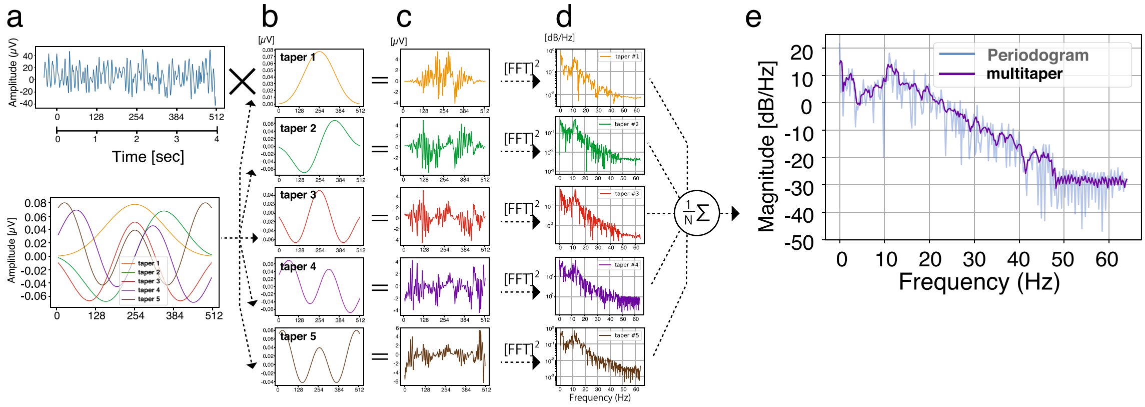


Fig. 7 Power spectral analysis using the adaptation of the multitaper method. **a** EEG wave (23-year-old male, under general anesthesia with sevoflurane inhalation, BIS = 55). Sampling frequency 128 Hz, 512 data points for 4 s. **b** Discrete prolate spheroid sequence (DPSS, Slepian sequence). **c** EEG waves multiplied by five Slepian sequence taper functions. **d** Each FFT result using the adaptation of each taper. **e** Power spectral density (solid red line) obtained by adding and averaging the FFT results from tapers, compared with that derived from Hann window processing (solid blue line)

two-dimensional scatter plot with a time axis and frequency axis, which displays the frequency band power at each data point as a color map. In this study, an EEG with a duration of 64 s (8,192 points) was analyzed at the three anesthesia stages under inhalational general anesthesia with sevoflurane (deep anesthesia, BIS value = 55; arousal process, BIS value = 70; BIS value at awakening time = 95) (Fig. 8a). The EEG segment was analyzed every 4 s (512 points) with a 50% overlap rate (31 analysis segments) (Fig. 8b), and the following three processing methods were used to create the spectrogram:

Method 1) Periodogram method with a Blackman window, adapted using *Matplotlib*'s *spectrogram()* function (Fig. 8c).

Method 2) Time–frequency reallocation method, using *tfr_spec()* from the library *libtfr*, is a method that uses the instantaneous frequency and phase values in a spectrogram to 'deconvolve' the image, and can yield substantially sharper spectrograms with better signal–noise resolution than conventional windowed spectrograms (Fig. 8d) [19, 20].

Method 3) Obtained by the MT method using the *libtfr* library's *mfft_dpss*() function (Fig. 8e).

For the spectrogram color map, the *Matplotlib* function was used in (1), and the display.spec s*how*() function from the acoustic library *librosa* was used in (2) and (3).

At the deep anesthesia stage, the PSD is characterized by peaks around 12 Hz and slow-wave power below 5 Hz. For the periodogram (method 1), the noise mentioned in Sect. 3 is mixed in the frequency band between the peaks around 12 Hz and around 5 Hz or less, which blurs the signal's overall characteristics (Fig. 8c). By contrast, for the time–frequency reallocation method (Fig. 8d), peaks around 12 Hz and below 5 Hz are displayed relatively sharply. For the MT method (Fig. 8e), the characteristic frequency bands are displayed clearly.

In the stage from anesthesia to awakening, 12-Hz-characteristic frequency band peaks almost fade out for both the periodogram and MT methods, but the higher gamma-wave region frequency band around 30 Hz increases the power. At the awakening stage, the power in the characteristic frequency band around 12 Hz disappears and is replaced by a uniform distribution up to the gamma-wave region exceeding 40 Hz; that is, in the spectrogram produced by the MT method, the effects of spectral leakage and noise are suppressed, and the characteristic frequency band can be visualized clearly, particularly during the deep stage of general anesthesia. In this respect, the spectral analysis and spectrograms produced by the MT method are superior for monitoring the characteristic changes in the EEG frequency band over time during general anesthesia.

7 Summary

The basics of frontal EEG analysis that focus on the spectrum and spectrogram during general anesthesia were demonstrated using *Python* programming in the *Jupyter Notebook* environment. Spectral leakage caused by the unnatural edge treatment of the analysis windows; window functions, such as Hann, Hamming, and Blackman; and the advantage of the MT method were explained using examples of graphs produced using *Python* programming under *Jupyter Notebook*. These software techniques are useful, and probably essential, for constructing an effective spectrum and spectrogram to enable the analysis of the relationship between the depth of anesthesia and EEG frequency analysis in general anesthesia.

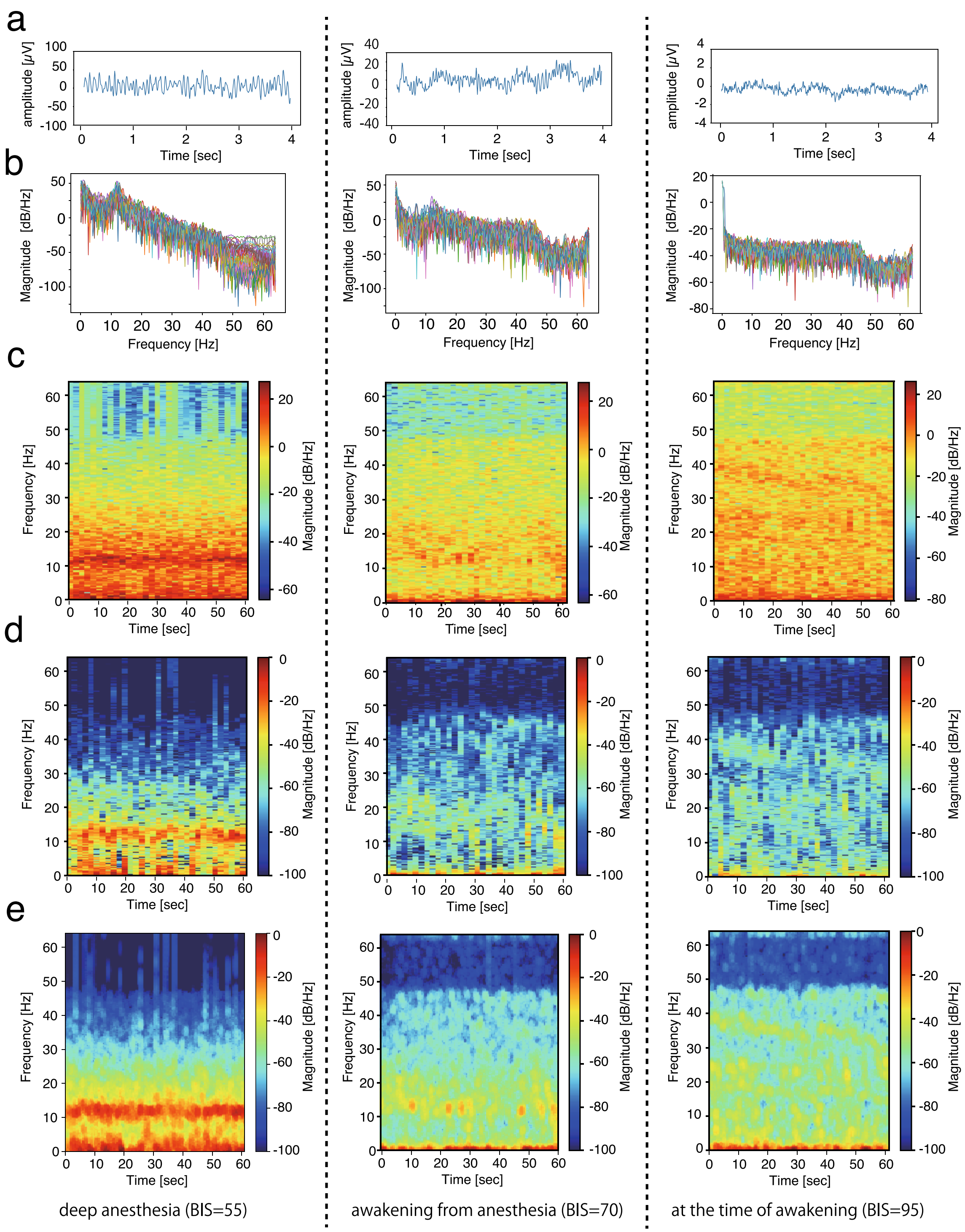


Fig. 8 EEG spectrogram analysis. Adaptation to EEG during general anesthesia with sevoflurane (23-year-old male, urachal residual excision, three stages during general anesthesia by inhalation of sevoflurane). Deep anesthesia (left), BIS value = 55; during awakening from anesthesia (center), BIS value = 70; at time of awakening (right), BIS value = 95. **a** EEG wave (first segment, 4 s, 512 data points). **b** Power spectral density. Thirty-one analysis segments displayed in the same view. **c** Spectrogram: periodogram (Blackman window) using *Matplotlib*'s *spectrogram()* function. **d** Spectrogram: time–frequency reassignment method using the *libtfr* library's *tfr_spec()* function. **e** Spectrogram: multitaper method using the *libtfr* library's *mfft_dpss()* function

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s10877-021-00771-4.

Acknowledgements We thank Edanz (https://jp.edanz.com/ac) for editing a draft of this manuscript.

Author contributions TS conducted the study, data collection, data analysis, and manuscript preparation. TY and YO helped revise the manuscript. All authors gave final approval of the submitted manuscript.

Funding We do not have any financial or other interest in any product mentioned in this manuscript.

Data availability The sample data and programming code using the analysis in this paper are available as supplementary data.

Code availability The supplementary archive file (spectral_analysis.zip) contains all the *Python* programming code files (.ipynb files of *Jupyter Notebook*, and the PDF print files) for the EEG analyses described in this paper: (1) EDF2rawEEG.pdf: *Python* code that transforms the EEG in EDF format to microvolt data. (2) eeg_bis_10_20_2.tsv: Sample EEG data used in the following spectral analyses. (3) spectral_analysis_1.pdf and spectral_analysis_1.ipynb: *Python* code for the spectral analysis of the EEG data using sine and cosine waves, and the ARIMA model. (4) spectral_analysis_2_deep_anesth.pdf and spectral_analysis_2_deep_anesth.ipynb: *Python* code for the spectral analysis of the EEG data at the phase of deep general anesthesia. (5) spectral_analysis_3_before_emeregence.pdf and spectral_analysis_3_before_emeregence.ipynb: *Python* code for the spectral analysis of the EEG data at the phase before emergence from general anesthesia. (6) spectral_analysis_4_emergence.pdf and spectral_analysis_4_emergence.ipynb: *Python* code for the spectral analysis of EEG data at the phase of emergence from general anesthesia.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval The EEG data used as examples for spectral analyses in this review were obtained from an anesthetized patient under ethical approval (No. ERB-C-1074-2) by the Institutional Review Board for Human Experiments at the Kyoto Prefectural University of Medicine (IRB of KPUM).

Informed consent For this non-interventional and noninvasive retrospective observational study, informed patient consent was waived by the IRB of KPUM; patients were provided with an opt-out option, of which they were notified in the preoperative anesthesia clinic.

Research involving human and animal rights The EEG data used as an example for the data analysis in this review were approved and the requirement for written informed consent was waived by the institutional review board of KPUM.

References

1. Fernández-Candil JL, Terradas SP, Barriuso EV, García LM, Cogollo MG, Gallego LG. Predicting unconsciousness after propofol administration: qCON, BIS, and ALPHA band frequency power. J Clin Monit Comput. 2021;35:723–9. https://doi.org/10.1007/s10877-020-00528-5.
2. Martín-Mateos I, Méndez Pérez JA, Reboso Morales JA, Gómez-González JF. Adaptive pharmacokinetic and pharmacodynamic modelling to predict propofol effect using BIS-guided anesthesia. Comput Biol Med. 2016;75:173–80. https://doi.org/10.1016/j.compbiomed.2016.06.007.
3. Alfonso-Pérez G, Méndez-Pérez JA, Torres-Álvarez ST, Morales JAR, Fragoso AML. Modelling the PSI response in general anesthesia. J Clin Monit Comput. 2010. https://doi.org/10.1007/s10877-020-00558-z.
4. Ching S, Cimenser A, Purdon PL, Brown EN, Kopell NJ. Thalamocortical model for a propofol-induced alpha-rhythm associated with loss of consciousness. Proc Natl Acad Sci USA. 2010;107:22665–70. https://doi.org/10.1073/pnas.1017069108.
5. Supp GG, Siegel M, Hipp JF, Engel AK. Cortical hypersynchrony predicts breakdown of sensory processing during loss of consciousness. Curr Biol. 2011;21:1988–93. https://doi.org/10.1016/j.cub.2011.10.017.
6. Flores FJ, Hartnack KE, Fath AB, Kim SE, Wilson M, Brown EN, Purdon PL. Thalamocortical synchronization during induction and emergence from propofol-induced unconsciousness. Proc Natl Acad Sci USA. 2017;114:E6660–8. https://doi.org/10.1073/pnas.1700148114.
7. Purdon PL, Sampson A, Pavone KJ, Brown EN. Clinical electroencephalography for anesthesiologists: part I: background and basic signatures. Anesthesiology. 2015;123:937–60. https://doi.org/10.1097/ALN.0000000000000841.
8. Sawa T. EEG Analyszer. ver 54_GP. Science to Medicine. http://anesth-kpum.org/blog_ts/?p=3169 (2020). Accessed 6 Nov 2020.
9. Hayase K, Kainuma A, Akiyama K, Kinoshita M, Shibasaki M, Sawa T. Poincaré plot area of gamma-Band EEG as a measure of emergence from inhalation general anesthesia. Front Physiol. 2021;12: 627088. https://doi.org/10.3389/fphys.2021.627088.
10. Fourier transform. Wikipedia. https://en.wikipedia.org/wiki/Fourier_transform (2021). Accessed 29 Sept 2021.
11. Discrete Fourier Transform (numpy.fft). NumPy. NumPy v1.21 Manual**.** The NumPy community. https://numpy.org/doc/stable/reference/routines.fft.html (2021). Accessed 29 Sept 2021.
12. Kemp B, Värri A, Rosa AC, Nielsen KD, Gade J. A simple format for exchange of digitized polygraphic recordings. Electroencephalogr Clin Neurophysiol. 1992;82:391–3. https://doi.org/10.1016/0013-4694(92)90009-7.
13. Kemp B, Olivan J. European data format “plus” (EDF+), an EDF alike standard format for the exchange of physiological data. Clin Neurophysiol. 2003;114:1755–61. https://doi.org/10.1016/s1388-2457(03)00123-8.
14. von Dincklage F, Jurth C, Schneider G, Garcia PS, Kreuzer M. Technical considerations when using the EEG export of the SEDLine Root device. J Clin Monit Comput. 2020. https://doi.org/10.1007/s10877-020-00578-9.
15. Thomson DJ. Spectrum estimation and harmonic analysis. Proc IEEE. 1982;70:1055–96. https://doi.org/10.1109/PROC.1982.12433.
16. Babadi B, Brown EN. A review of multitaper spectral analysis. IEEE Trans Biomed Eng. 2014;61:1555–64. https://doi.org/10.1109/TBME.2014.2311996.
17. Kim S-E, Behr MK, Ba D, Brown EN. State-space multitaper time-frequency analysis. Proc Natl Acad Sci. 2018;115:E5–14. https://doi.org/10.1073/pnas.1702877115.
18. Slepian D. Prolate spheroidal wave functions, fourier analysis, and uncertainty - V: the discrete case. Bell Syst Techn J. 1978;57:1371–430. https://doi.org/10.1002/j.1538-7305.1978.tb02104.x.

19. Reassignment method. Wikipedia. https://en.wikipedia.org/wiki/Reassignment_method Accessed 6 Nov 2020.
20. Meliza D. libtfr. fast multitaper conventional and reassignment spectrograms. https://github.com/melizalab/libtfr (2021). Accessed 17 Mar 2021.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.