

**Griffith School of Engineering
Griffith University**

7506ENG – Industry Affiliates Program

ARIMA Modeling of Noise in Electromyography Signals

Waleed Umer, s5336317

10/17/2025, Trimester 2, 2025

**Griffith University
Dr.Faisal Mohd Yasin**

*A report submitted in partial fulfillment of the degree of Master of Professional Engineering
(Electronics)*

The copyright on this report is held by the author. Permission has been granted to Griffith University to keep a reference copy of this report.



EXECUTIVE SUMMARY

The project formalizes the primary workflow to be used in the detection and modeling of noise in electromyography (EMG) signals- a sector where no prior studies directly modeled the noise component itself. Multi-channel data sets (under 4 and 12 channels) are consolidated into a common format - sorted by subject, gesture, trial, channel, and normalized to constant length and constant sample rates so that they can be easily compared across all sources. Noise extraction is carried out in ordered steps.

To start with, a wide 50/60 Hz-power line artefact (as well as its harmonics) is eliminated by means of sinusoidal regression, which does not affect its nearby frequency content, which might reflect actual muscle activity. Second, short window RMS envelopes are used to isolate rest periods with a strong threshold, together with a rule of a minimum duration and gap fill-in to avoid over-segmentation. In either case, this produces (i) a noisy signal-only version of the original data, and (ii) a rest-noise (inter-gesture noise) sequence to be modeled. Simple diagnostics of the Welch power spectrums and time domain visual verifications ensure that the line artefacts are minimized and the resulting signal behaves as random noise before the actual model mining. Simple interpretable ARMA/ARIMA models are then used to describe the extracted noise. When it is needed that the differences be used are specified, by standard statistical tests (ADF, KPSS, PP); and the order of the model (p, d, q) is selected by goodness-of-fit tests (residual ACF/PACF and Ljung-Box). The low-order models, usually ARIMA (1,0,3) or ARIMA (2,0,3), can describe the short-memory nature of EMG noise across datasets. A methodical analysis with the reduction of the size of data ($N, N/2, N/4$, etc.) reveals that one can reliably identify the data with small samples, which can be of practical concern to designing an experiment and determining the level of data necessary.

This uniform workflow can be clarified and reproduced: all the parameters are written in material units (seconds, hertz), and all the decisions are documented according to each dataset and each trial; there is no double impact of outcome differences due to data characteristics and not to processing bias. In general, the method combines uniform segmentation, noise removal by rest, stringent verification, and statistically valid ARIMA-modeling to provide credible accounts of EMG noise. Outside of characterization, this modeling structure provides an opportunity to act as a possible alternative to conventional filter-based denoising, which offers a data-driven means of reconstruction and removal of noise, and still preserves the actual activity of the muscle. It further provides the base to extensions like multichannel adaptive modeling, intelligent EMG noise removal in biomedical and prosthetic usage.

ACKNOWLEDGEMENTS

I wish to express my profound gratitude to Dr. Faisal Mohd Yasin for his supervision, rigorous guidance, and constructive critique throughout the course of this project. His expertise and steady counsel were indispensable to the development of this thesis. I also extend sincere thanks to Mr. Ghulam Ali for his consistent support, insightful discussions, and practical suggestions that strengthened both the analysis and presentation of results. Finally, I acknowledge the encouragement and assistance of all who contributed, directly or indirectly, to the successful completion of this work.

TABLE OF CONTENTS

EXECUTIVE SUMMARY	2
ACKNOWLEDGEMENTS	3
TABLE OF CONTENTS	4
LIST OF FIGURES	6
LIST OF TABLES	7
1 INTRODUCTION	8
1.1 Project background	9
1.2 Research Motivation.....	9
1.3 Research Aims and Objectives	10
1.4 Thesis Structure.....	11
2 LITERATURE REVIEW	12
2.1 Fundamentals of electromyography signals: physiology, instrumentation, and bandwidth	12
2.1.1 Overview of EMG Artefacts and Noise Sources	13
2.1.2 Removal of power-line interference by model-based regression	14
2.1.3 Noise acquisition from quiet segments via RMS + robust thresholds	14
2.1.4 Spectral diagnostics using Welch power spectral density	15
2.1.5 Stationarity assessment in short windows.....	16
2.1.6 Autoregressive integrated moving-average (ARIMA)	16
2.1.7 ARMA/ARIMA modeling and adequacy diagnostics	17
2.2 Alternatives not used as core stages.....	18
2.3 Cross-dataset considerations	19
2.4 Threats to Validity & Mitigations.....	20
3 DESIGN METHODOLOGY & DATASET PROTOCOLS	20
3.1 Methodological overview	21
3.2 Datasets and acquisition context	22
3.3 Raw Data Segmentation.....	25
3.4 Noise Data Acquisition & Verification	26
3.4.1 Overview.....	26
3.4.2 Stage 1 — Mains-hum regression (deterministic component)	27
3.4.3 Stage 2 — Quiet-segment noise acquisition (stochastic component).....	28
3.4.4 Verification that the harvested signal is “noise”	29

3.4.5	Parameter harmonization across datasets.....	30
3.4.6	Quality assurance	30
3.5	Testing stationarity and modeling noise with the ARMA/ARIMA method.....	31
3.5.1	Inputs and progressive subsampling	31
3.5.2	Preparing the series	31
3.5.3	Stationarity checks and differencing.....	31
3.5.4	Selecting an ARMA/ARIMA model	32
3.5.5	Is the model good enough? (adequacy)	32
3.5.6	How outcomes are labelled.....	32
3.5.7	What changes with shorter data?	33
3.5.8	ARIMA model-generated noise vs. measured noise.....	33
3.6	Validity of the Approach.....	34
3.6.1	Alignment with accepted practice and study goals.....	34
3.6.2	Validity of segmentation and standardization.....	34
3.6.3	Validity of mains-hum regression instead of fixed notches.....	35
3.6.4	Validity of quiet-segment noise acquisition (empirical noise)	35
3.6.5	Validity of spectral diagnostics (Welch PSD and simple time-domain checks) .	35
3.6.6	Validity of stationarity testing and differencing policy	36
3.6.7	Validity of ARMA/ARIMA identification with small-sample safeguards.....	36
3.6.8	Validity of progressive subsampling (N, N/2, ...).....	36
3.6.9	Validity of model-based subtraction (demonstration)	37
3.6.10	Cross-dataset harmonization and reproducibility	37
3.6.11	Threats to validity and mitigations	37
3.6.12	Overall assessment.....	38
3.7	Risks, limitations, and mitigations	38
4	RESULTS AND ANALYSIS	39
4.1	Overview.....	39
4.2	Raw Data Segmentation Results	39
4.2.1	Dataset I (DI)	39
4.2.2	Dataset II (DII).....	40
4.2.3	Dataset III (DIII).....	41
4.2.4	Dataset IV (DIV).....	42
4.3	Noise Segmentation results (DI–DIV).....	43
4.4	Electromyography Signal Noise Composition	48

4.5 Stationarity Tests.....	50
4.6 ARIMA Modeling Results.....	51
4.6.1 Dataset-I ARIMA model.....	51
4.6.2 Dataset-II ARIMA model	54
4.6.3 Dataset-III ARIMA model.....	58
4.6.4 Dataset-IV ARIMA model.....	63
4.7 Key Findings	68
5 CONCLUSION	69
REFERENCES.....	71
APPENDIX A : DATASET SUMMARIES.....	74
APPENDIX B : ADDITIONAL RESULTS.....	75
APPENDIX C : CODE REFERENCES	77

LIST OF FIGURES

Figure 1. Decomposition of EMG signal into constituent components.....	8
Figure 2. Power-line interference regression concept.....	13
Figure 3. Common artefacts in EMG and their spectral characteristics	14
Figure 4. Project Methodology Flowchart	21
Figure 5. Forearm electrode layout (FORS Dataset)	25
Figure 6. End-to-end method with inputs/outputs and diagnostics per stage	26
Figure 7. PSD overlay before and after mains regression	28
Figure 8. RMS + MAD gating and quiet segment extraction	29
Figure 9. Residual ACF for accepted models	33
Figure 10. Segmentation results for DI.....	40
Figure 11. Segmentation results for DII	41
Figure 12. Segmentation results for DIII	42
Figure 13. Segmentation results for DIV	43
Figure 14. Noise signals for the rest gesture in DI	44
Figure 15. Noise signals for the rest gesture in DII	45
Figure 16. Noise signals for the rest gesture in DIII.....	46
Figure 17. Noise signals for the rest gesture in DIV.....	47
Figure 18. Noise vs Rest Extracted Noise	50
Figure 19. Original noise vs. ARIMA fitted DI.....	51

Figure 20. ARIMA Residuals Plot and ACF Residuals Plot DI.....	52
Figure 21. ARIMA noise modeling application	53
Figure 22. ACF/PACF for Dataset II.....	54
Figure 23. ARIMA adequacy panel for Dataset II.....	56
Figure 24. ARIMA application.....	57
Figure 25. ARIMA model diagnostics T1- DIII	59
Figure 26. ARIMA model diagnostics T2- DIII	60
Figure 27. ARIMA model diagnostics T3-DIII	61
Figure 28. . ARIMA noise modeling application DIII.....	62
Figure 29. ACF for Trial 1 - DIV	64
Figure 30. Original versus fitted ARIMA model output for Dataset IV.....	64
Figure 31.Original versus fitted ARIMA model output for Trial 05- Dataset IV.....	65
Figure 32. Residual whiteness Trial 5- DIV	66
Figure 33. ARIMA-based noise modeling application DIV	67

LIST OF TABLES

Table 1. Comparison of Noise acquisition and modeling methods	15
Table 2. Dataset Summary (D1–D4)	18
Table 3. PSD (Power Spectral Density) metrics.....	31
Table 4. Noise Composition Across D1	48
Table 5. Welch-PSD diagnostics on “rest” noise DI	49
Table 6. Welch-PSD diagnostics	49
Table 7. Summary of stationarity outcomes for selected cases from DI rest gesture.....	50
Table 8. ARIMA Summary for channel 1 across 5 trials (Dataset I)	53
Table 9.ARIMA Root Mean Square Error of Residuals	68
Table 10. Dataset III Detailed Acquisition Summary.....	74
Table 11. Dataset IV Detailed Acquisition Summary	74
Table 12. ARIMA modelling results across progressive subsampling cases (Channel 1)-DI.	75
Table 13. ARIMA modeling across halving cases (Dataset II : Ch1)	75
Table 14. ARIMA modeling results (Dataset III : Ch1).....	76
Table 15. .ARIMA modeling results (Dataset IV : Ch1).....	76

1 INTRODUCTION

Electromyography signals are the superposition of motor unit action potentials of skin measured by difference electrodes and low noise amplification [1], [2]. These signals are in practice caused by a cascade of biophysical filtering, volume conductor, electrode-skin interface, pre-amp filter, anti-alias filter, and can be sampled in the range of one kilohertz to retain diagnostically useful bandwidth [1], [3]. Since the recorded waveform is of small amplitude and is recorded under normal conditions, it is often corrupted by various artefacts: narrowband interference due to power lines 50/60 Hz and harmonic of that interference; low frequency base effects and motion drift due to electrode micro movements and impedance variations; broadband instrumentation noise due to the front end; and cross talk due to other adjacent muscles [1], [3].

The traditional techniques follow the idea of eliminating such artefacts using approximately determined filters and defining ad hoc thresholds in order to analyze physiology [3]. This thesis will be different: first, we isolate it and characterize noise and only model it explicitly using model-based noise handling concepts of the larger signal processing literature [4]. The main assumption is that noise is not an adverse event that should be reduced but rather a process that can be learned and removed in a compensated manner. The previous EMG studies have not reported this point of view, where noise has generally been regarded as an undesirable artefact instead of a dynamic process, which could be statistically described and modeled. Figure 1 depicts the breakdown of the electromyography signals into their portions of motor unit action potentials(MUAPs).

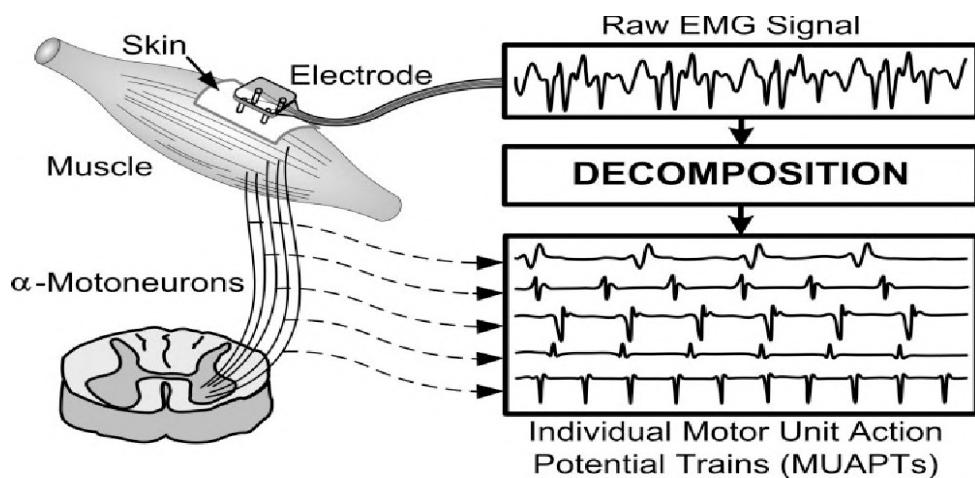


Figure 1. Outline of the decomposition of the surface EMG signal into its constituent motor unit action potentials [5]

1.1 Project background

Electromyography signals have long been modeled with linear stochastic processes for purposes such as activation detection, prosthesis control, and spectral parameterization [6]. Many works approximate electromyography short windows as autoregressive (AR), moving average (MA), or autoregressive moving average (ARMA) processes, and some of the works assume non-stationary cases modeled using the autoregressive integrated moving average (ARIMA) model [6], [7], [8]. Such attempts are normally focused on the physiological aspect, where artefacts are considered exogenous contamination that should be filtered [1], [3]. Nonetheless, there are two facts about that. To begin with, power line interference and its harmonics are in and may overlap the physiological passband; violent notching may distort the spectral shoulders flanking pulses and blur information about the firing of motor units [1], [3]. Second, the baseline/motion drift. Second, when real contractions are slow modulated, high-pass filters robust enough to remove the drift can also remove valid low-frequency information [1], [3]. Consequently, completely filter-centric approaches usually have a dilemma choice either residual artefacts contaminate subsequent features or collateral distortion is induced. The process of noise modeling is a principled alternative. When we can obtain silent intervals where muscles are not moving or touching each other, we get empirical realizations of the noise which are conditioned by the same hardware, stimulus environment, and electrode location as are the task data [1]. Based on these segments, it is possible to estimate compact, low-order ARMA/ARIMA models (with roughly white innovations), create noise surrogates that are equal in their variance and spectrum, and remove them from electromyography along with noise signals [7]. The benefit is two-fold: (i) subtraction is model-based and not frequency surgical, collateral damage decreases [4]; (ii) the fitted models can give quantitative measures of quality (innovation variance, AICc, whiteness tests), rendering the quality of acquisition reproducible [7]. Despite existing literature demonstrating the application of AR, ARMA, and ARIMA models to electromyography (EMG) signals [6], [8], there appears to be limited rigorous work that specifically models the noise component itself using ARIMA. Furthermore, few studies have demonstrated the subtraction of modeled noise from held-out signals or its effectiveness in *in vivo* conditions. This gap is addressed by the present thesis.

1.2 Research Motivation

EMG recordings usually have undesirable skin-electrode contacts, power-line hum, and equipment junk. These artefacts lie within the same frequency range as actual muscle activity,

and therefore simple on / off filters either retain too much noise, or chop up the signal of interest[1], [3].

The intervals between rests in the data provide a clear view of the noise of the specific sensors and configuration. The rest noise can be modeled based on such true data, and this approach enables the team to model it clearly and understandably, as well as verify the model using simple residual tests [6], [7].

In case the modeled noise is similar to the actual noise, it could be removed in EMG + noise recordings. This, in practice, must imply reduce 50/60 Hz hum, reduced slow drift, and reduced overall variance, a method that is easy to read compared to heavy filtering, and is a foundation of regular quality checks [4], [6], [7].

In addition to cleaning signals, a good noise model is also handy in constructing realistic test data. It assists in modeling hard conditions of recording and stress-testing decoding techniques without a fake muscle activity was invented [7].

1.3 Research Aims and Objectives

This project aims to do three things. First, it aims to reliably collect the background noise present in multi-channel forearm electromyography signals by using simple, rest-based gating so the extracted signal truly reflects noise rather than muscle activity. Second, it aims to describe that noise with ARIMA models, checking that the leftover errors (residuals) are small and uncorrelated to show the model is a good fit. Third, it aims to use the learned ARIMA noise to subtract contamination from EMG-plus-noise recordings, reducing interference without distorting the underlying muscle signal.

Some of the key project objectives are as follows:

1. Find and collect “rest” parts of each channel—times when there is no muscle activity—using a simple RMS (root mean square) method, and stitch those rest samples together to make a clean noise track per channel.
2. Describe what that noise looks like in frequency terms: check how much power sits at 50/60 Hz (and its multiples), how much is in the very low vs. high bands, and how flat or “white” the spectrum appears.

3. Check whether the noise stays reasonably steady over short periods and, when needed, lightly differ it to remove slow drifts, then fit small ARIMA models to each channel's rest-only noise.
4. Make sure each model is sensible: the residuals after modeling should be small and show no obvious repeating patterns, and the chosen model should be as simple as possible while still fitting well.
5. Use the fitted models to generate noise and subtract it from recordings that contain both EMG and noise; judge improvement by lower variance, better frequency metrics, and, when a clean reference exists, lower error.
6. Record all settings and list the per-channel model formulas so others can repeat the work and reuse the results.

1.4 Thesis Structure

The thesis is outlined as follows:

- **Chapter 2** – Literature Review. Background relevant to handling noise in electromyography signals.
- **Chapter 3** – Datasets & Protocol (Methods Included). Datasets, acquisition settings, channel layout, and all methods used in this work (mains hum regression, rest-segment extraction, PSD metrics, and ARMA/ARIMA modeling), with parameters and implementation details.
- **Chapter 4** – Results. Segmentation, noise acquisition, and modeling results with analysis.
- **Chapter 5** – Conclusions. Key findings, limitations, and directions for future work.

2 LITERATURE REVIEW

This chapter brings together the background concepts needed to understand ARIMA-based noise modeling for electromyography (EMG) signals, where noise is directly collected, analyzed, and modeled. It begins by describing the main sources of unwanted signals in EMG recordings, such as power-line hum and movement-related artefacts. It then outlines, in simple terms, the methods used in this project: removing mains interference by fitting and subtracting sinusoids, identifying rest periods using a moving energy measure with set thresholds, checking whether the isolated portions truly behave like noise by examining their frequency content, and applying small ARIMA models on these rest segments. The chapter also briefly compares other common approaches, including independent component analysis, empirical mode decomposition, and wavelet-based techniques. Finally, it highlights frequent mistakes that can lead to unreliable conclusions and suggests straightforward practices to make the analysis process easier to repeat and verify.

2.1 Fundamentals of electromyography signals: physiology, instrumentation, and bandwidth

The electromyography (EMG) signals are formed when the tiny electrical signals emitted by muscle fibers are added and spread across the tissues in the body to the skin electrodes. The signal recorded varies with various factors: the electric activity level of the various motor units and their firing behavior, the conduction of the signal along the muscle fibres, and the filtering of the signal by skin and fat layers before it is sent on to the electrodes. The electrode shape and quality also vary depending on the electrode design, and the distance between the electrodes and the recording configuration, whether the difference between points (differential) or a single reference point (monopolar). A combination of these factors defines the extent to which the EMG signal will reflect the actual activity of the muscle, as well as the degree of unwanted interference in the signal. The important parameters to prevent artefact contamination are amplifier input impedance, common mode rejection ratio (CMRR), and anti-alias filtering. Merletti and Farina [1] provide canonical coverage, and the practical survey of the choices of the process is in Reaz et al. [3].

Common voluntary sEMG (surface electromyography) energy is focused in 20-450 Hz, and dependent on the muscle and task. Baseline/motion artefacts are most affected at very low frequencies (below 20 Hz), whereas frequency bands at 50/60 Hz overlap power line interference (PLI) [1], [3].

2.1.1 Overview of EMG Artefacts and Noise Sources

Electromyography recordings are routinely contaminated by:

- Power-line interference (PLI) near 50/60 Hz and harmonics
- Baseline/motion drift (<20–30 Hz) driven by electrode-skin impedance fluctuations and limb movement
- Broadband instrumentation noise (front-end/preamplifier and quantisation)
- Cross-talk from nearby muscles [1], [3]. These artefacts overlap physiological content in both time and frequency, so purely filter-centric methods risk either residual contamination or distortion of the EMG of interest [1], [3]. Noise removal method from EMG signal separates the problem: first, acquire and validate the noise, then model it and subtract it in a controlled way.
- Signals from nearby muscles that the electrodes also pick up. These can show up across several channels at once. They aren't pure "noise," but they do interfere when the goal is to analyze one specific muscle.

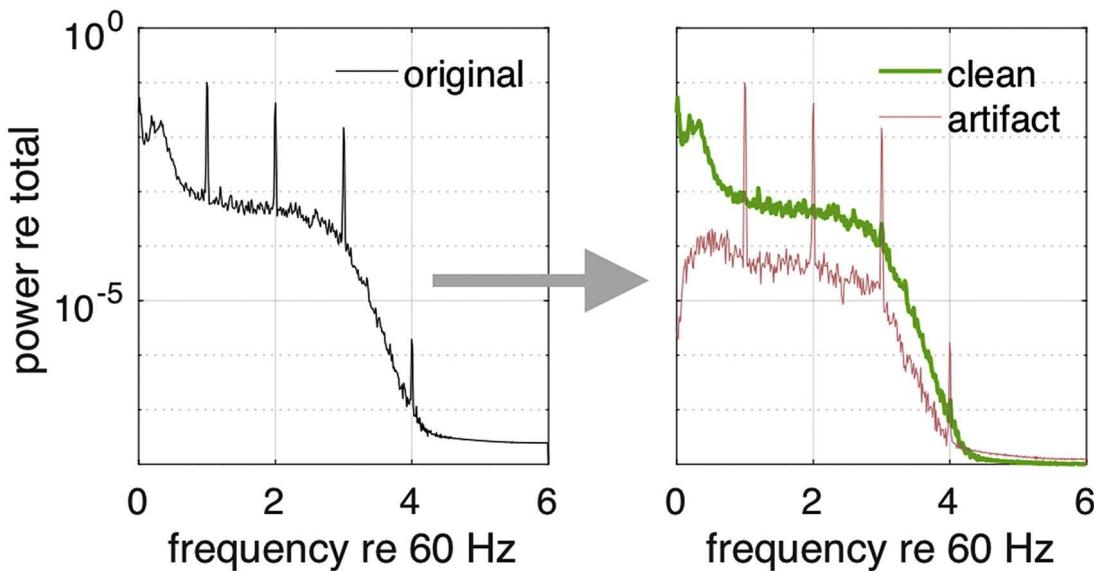


Figure 2. Illustration of power line artefact removal using the ZapLine method [9]

In Figure 2 left panel shows the power spectral density of the original signal with distinct harmonic peaks around the power-line frequency. The right panel demonstrates the separation of the cleaned signal (green) and the extracted artefact component (red), showing effective suppression of line-related interference while preserving the underlying physiological spectrum.

2.1.2 Removal of power-line interference by model-based regression

Motivation: Fixed “notch” filters are common for removing mains hum, but they can cut into nearby muscle signal, create ringing when the hum drifts in phase, and struggle with multiple harmonics. When it is important to preserve the surrounding frequencies, a safer choice is to model the line noise and subtract it (or interpolate the narrow spectral lines), as is standard in EEG work. [9].

Method. The hum is modeled parametrically as a finite sum of sinusoids at the fundamental $f_0 \in \{50,60\}$ Hz and K harmonics:

$$\hat{h}(t) = \sum_{k=1}^K [a_k \cos(2\pi k f_0 t) + b_k \sin(2\pi k f_0 t)] \quad (1)$$

Coefficients $\{a_k, b_k\}$ are estimated by least squares over sliding windows to accommodate slow amplitude/phase drift; the fitted $\hat{h}(t)$ is subtracted channel-wise [9]. Compared with rigid notching, sinusoid regression preserves spectral shoulders adjacent to the line frequency.

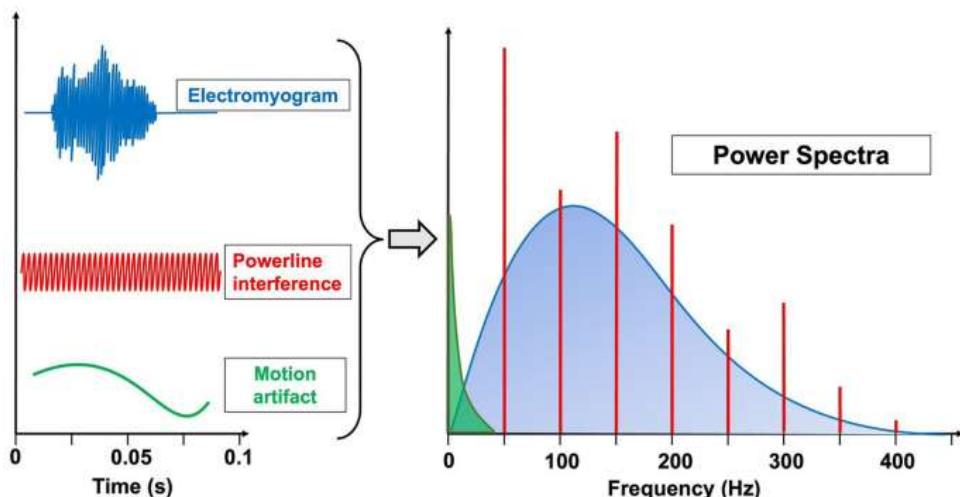


Figure 3. Illustration of common artefacts in electromyographic (EMG) recordings and their corresponding spectral characteristics[10]

2.1.3 Noise acquisition from quiet segments via RMS + robust thresholds

Instead of guessing what the noise looks like or creating it artificially, the study takes it directly from the data. It finds natural rest periods—times when muscle activity is minimal—on each channel, extracts those portions, and stitches them together. The result is a noise signal in the time domain that can be described, compared, and modeled [1], [3].

For a channel $x(t)$, the RMS envelope over a window of W samples is

$$e(t) = \sqrt{\frac{1}{W} \sum_{i=0}^{W-1} x^2(t-i)} \quad (2)$$

A robust threshold $\tau = \text{median}(e) + k \text{MAD}(e)$ (*typically* $k \in [1.5, 2.0]$) is used to delineate quiet regions; the median absolute deviation (MAD) confers resistance to outliers and occasional bursts [11]. Minimum duration and gap-fill rules are enforced to prevent fragmentation. Two outputs are retained: (i) a masked trace of identical length to the original signal, and (ii) a concatenated sequence consisting only of quiet segments for subsequent modeling.

The RMS window length W and the threshold multiplier k control what gets labeled as “rest.” A larger W smooths the envelope and can merge short bursts into the baseline; a smaller W is more sensitive to brief transients. A higher k sets a looser threshold (cleaner segments but fewer of them), while a lower k captures more data at the risk of including mild activity.

2.1.4 Spectral diagnostics using Welch power spectral density

For short, nearly steady segments, a single spectrum estimate can be noisy. Welch’s method fixes this by slicing the data into overlapping pieces, applying a window to each piece, computing their spectra, and then averaging the results. This averaging lowers random variation while keeping bias small, so the final power–frequency curve is much more stable and trustworthy [12]. At ~ 1 kHz sampling, segment lengths of 512–1024 samples with 50% overlap are commonly used, and a Hann window is recommended to control sidelobes [13].

From each Welch power spectrum, a small set of simple checks is reported to describe the noise. A line-ratio measures how much power sits at the mains frequency (50/60 Hz) and its harmonics—higher values mean more leftover hum. Low- and high-frequency shares track how much energy lies below about 20–30 Hz (slow baseline or motion) and above a chosen upper band (e.g., ≥ 150 Hz), which together show whether the spectrum tilts toward slow drift or higher-frequency content. The spectral flatness measure compares the geometric and arithmetic means of the spectrum; a higher value looks more “white” and noise-like [14]. Two quick time-domain screens complement the spectrum: a spike index counts brief outliers using

robust z-scores, and an ECG-leakage cue looks for 5–40 Hz power with a repeating pattern in the autocorrelation that would suggest heart-related contamination.

Together, these metrics validate that the harvested sequence is artefactual and guide parameter tuning (e.g., hum harmonics, gating thresholds).

2.1.5 Stationarity assessment in short windows

Electromyography noise is treated as locally stationary over short windows (hundreds of samples), rather than over entire trials. Three complementary tests are standard:

- ADF (Dickey–Fuller). Tests a unit-root null in autoregressive regressions; small p-values reject non-stationarity [15].
- Phillips–Perron (PP). Tests the unit-root null with nonparametric corrections for serial correlation and heteroskedasticity [16].
- KPSS. Tests the stationarity null; large p-values support level-stationarity [17].

Consistent outcomes (ADF/PP small, KPSS large) support ARMA ($d=0$) modeling; mixed outcomes suggest ARIMA ($d=1$).

2.1.6 Autoregressive integrated moving-average (ARIMA)

Autoregressive integrated moving-average (ARIMA) models describe a time series that becomes (locally) covariance-stationary once an appropriate number of differences has been taken. After differencing, the series is modeled as the sum of an autoregressive (AR) component—where current values depend linearly on a finite number of their own past values—and a moving-average (MA) component—where current values depend linearly on a finite number of past random shocks (innovations). The model is denoted ARIMA(p,d,q), where d is the number of differences applied, p is the AR order, and q is the MA order. In the backshift operator B , an ARIMA(p,d,q) for a scalar series x_t is written as

$$\phi(B)(1-B)^d x_t = \theta(B) \varepsilon_t, \varepsilon_t \sim WN(0, \sigma^2) \quad (3)$$

with $\phi(B) = 1 - \phi_1 B - \dots - \phi_p B^p$ and $\theta(B) = 1 + \theta_1 B + \dots + \theta_q B^q$ defining the AR and MA polynomials, respectively [13]. The differencing order d is chosen to remove low-frequency trends so that the transformed series is well-approximated by a short-memory

ARMA process; in practice, unit-root and stationarity tests (ADF, PP, KPSS) guide d selection [15], [17], [18]. ARIMA is widely used because it is parsimonious, interpretable, diagnostically testable (via residual whiteness and information criteria), and generative. The fitted ARIMA models can also be used to generate realistic artificial noise sequences that closely resemble real EMG noise. These simulated results provide a safe and repeatable way to test filtering and classification methods without requiring new data collection. [16]. In short, ARIMA is preferable when (i) the series is reasonably linear and (locally) stationary after modest differencing, (ii) dependence is short-range and well captured by low orders p,q (iii) there is no strong seasonality (or seasonality is handled separately), and (iv) small-sample bias is controlled by criteria such as AICc, which penalises extra parameters more heavily than AIC when N is limited [19]. Order identification can be conducted by constrained automatic procedures (e.g., Hyndman–Khandakar) that combine unit-root testing, stepwise search over low orders, and information-criterion ranking, followed by residual diagnostics to confirm adequacy [16], [20].

2.1.7 ARMA/ARIMA modeling and adequacy diagnostics

ARMA and ARIMA models are used to describe short-term dependencies in the data, with ARMA(p,q) applied when the series is already stable and ARIMA(p,d,q) when mild differencing is needed to make it stable [16]. These models are built under the condition that they remain mathematically stable and reversible, meaning the estimated coefficients keep the process well-behaved (roots outside the unit circle). The choice of model order aims to balance simplicity and accuracy. Since the noise segments are short, small-sample bias is a concern, so the corrected Akaike Information Criterion (AICc) is preferred over AIC to discourage adding unnecessary parameters [19]. In practice, a restricted automatic search similar to the approach of Hyndman and Khandakar [20] is used to explore only low model orders, followed by quick reasonableness checks such as whether the AR and MA terms have expected signs and magnitudes and whether the innovation variance is sensible.

Adequacy diagnostics (model behaviour) then confirm whether the chosen model genuinely captures the behaviour of the noise. These tests examine the **residuals**, or the difference between the model's prediction and the actual signal. If the residuals resemble random white noise—showing no visible correlation or pattern—it indicates that the model fits well. This is verified using residual autocorrelation plots (ACF and PACF) and the Ljung–Box test, typically at a lag around one-fifth of the data length ($m \approx N/5$) [21]. Supportive checks, such as residual

ADF and KPSS tests, can also be applied to confirm that no further structure remains. Passing these adequacy diagnostics means the extracted noise has been well represented by a compact ARMA/ARIMA process within the analyzed window.

Table 1. Comparison of Noise acquisition and modeling methods

Approach	Principle	Advantages	Limitations	Typical Parameters	Key References
Model-based power-line regression	Fits sinusoids at 50/60 Hz ± harmonics to remove deterministic hum.	Preserves nearby physiological bands; tunable; avoids fixed notches.	Requires harmonic count K selection; fails for non-sinusoidal sidebands.	$f_0 = 50/60 \text{ Hz}$; $K = 1-3$; window 0.5–1 s (50 % overlap).	[4], [9], [12]
RMS + MAD quiet-segment gating	Identifies low-activity “noise” intervals via RMS envelope and MAD-based thresholds.	Empirical noise estimation; robust to outliers; supports cross-channel logic.	AND-gating reduces yield; sensitive to k and window W.	$W \approx 100 \text{ ms}$; $k \approx 2-3$; AND vs OR gating.	[9], [21]
Welch Power Spectral Density diagnostics	Averages windowed periodograms to verify the spectral content of noise.	Low variance; interpretable metrics (LR, LF/HF, SFM).	Trade-off between window length and resolution.	512–1024 samples; 50 % overlap; Hann window.	[12], [13], [21]
Stationarity tests (ADF, PP, KPSS)	Determines if a series is level-stationary or needs differencing.	Complementary tests avoid false decisions; support ARMA/ARIMA choice.	Short segments increase variance; there may conflict between tests.	Window $\approx 0.5-1 \text{ s}$; p threshold 0.05.	[15]–[17]
ARIMA / ARMA noise modeling	Captures short-memory temporal dependence in noise series.	Parsimonious; diagnostically testable; supports simulation and forecasting.	Assumes linearity; fails for non-stationary or multi-sensor artefacts.	$p, q \leq 3$; $d = 0-1$; AICc for selection; Ljung–Box for whiteness.	[14], [16], [19]–[20]

2.2 Alternatives not used as core stages

While not core to the implemented method, the following methods remain valuable alternatives for datasets with stronger shared artefacts or pronounced non-stationarity:

- Independent Component Analysis (ICA): This method separates mixed signals from multiple channels by taking advantage of statistical differences between them (non-Gaussianity). Algorithms such as FastICA and JADE are commonly used. ICA works well when the noise or interference is shared across many channels, but it can be sensitive to incorrect assumptions about the data and may produce uncertain results if the signal order or mixing model is not accurate. [21].

- Empirical Mode Decomposition (EMD/EEMD): This method breaks down a complex signal into a set of simpler wave-like components called intrinsic mode functions (IMFs). The improved version, Ensemble EMD (EEMD), adds small amounts of noise and averages the results to prevent “mode mixing,” where different frequency components overlap. This increases the stability in the output, but it consumes more time to compute [22], [23].
- Included Shrinkage of maximal overlap DWT (MODWT). The technique employs wavelets to eliminate the undesired noise and still retains the time of the original signal. In comparison with ordinary wavelet transforms, MODWT circumvents the shifting problems that may arise in resampling, and it can handle the edge effect more successfully. It is, yet, computationally more complex than the simplest discrete wavelet transform (DWT) [24], [25].

2.3 Cross-dataset considerations

In case two or more datasets are to be different in terms of channel layouts, electrode placements, gesture protocols, and sampling rates, it is necessary to tune the electrode placement parameter and the remaining parameters so that they all match. When performing power-line hum removal, the correct mains frequency (50 Hz or 60 Hz) must first be verified. The amount of harmonics to be removed and the length of the fitting window should be selected to provide stable results, and the duration of re-fitting (applied per time window or trial) should also be set.

In the case of RMS-based rest detection, the signal energy window being considered should be about 100 milliseconds in duration, although this energy window will depend on the sampling rate. The multiplier, which decides what is considered rest, also needs to be tweaked to achieve the goal of ensuring that enough data is captured, and also making the noise segments clean since the background levels vary across different datasets.

In the case of power spectral density estimation by the Welch method, segment lengths should be 512-1024 samples with 50% fast forward overlap and a Hann window to produce smooth, constant spectra. The low- and high-frequency cutoffs applied to summary ratios must be reported in hertz to ensure transparency. In the case of the ARIMA model, both window sizes are to be reported using samples and seconds, and its order is kept to a minimum with a consistent selection criterion like AICc.

2.4 Threats to Validity & Mitigations

Several potential threats to validity were identified in the proposed methodology, along with corresponding mitigation strategies:

- **Mis-labelled quiet segments:** Small muscle activations can sometimes be too weak to pass the detection threshold, causing them to be incorrectly marked as rest. This can be reduced by requiring several channels to agree before marking a rest period (AND-gating), by slightly tightening the threshold value, and by setting a minimum duration so very short bursts are not misclassified.
- **Non-sinusoidal mains hum:** Electrical interference from power supplies and switching devices can create complex patterns in the signal that are not fully captured by simple sine-wave models. This issue can be reduced by including more harmonics in the model, updating the fit repeatedly over short windows, and using adaptive noise cancellation when a clean reference signal is available [4].
- **Welch-PSD bias–variance trade-off:** If the analysis segments are too short, the results can become unstable and unreliable, while overly long segments may break the assumption that the signal remains consistent over time. To avoid this, a balanced segment length (typically 512–1024 samples) is used, along with a Hann window and 50% overlap to ensure smoother and more dependable estimates. [13].
- **Over-differencing in ARIMA models:** Applying differencing when it is not needed can remove important low-frequency patterns that are part of the actual signal. To prevent this, the choice of model is guided by agreement between the ADF, PP, and KPSS tests for stationarity, with a preference for simpler ARMA models whenever the data allow.
- **Small-sample model selection bias.** The Akaike Information Criterion (AIC) tends to over-fit when applied to short time windows. Mitigation: The corrected version (AICc) is employed, and maximum autoregressive and moving-average orders are set to prevent over-fitting [19].

3 DESIGN METHODOLOGY & DATASET PROTOCOLS

This chapter presents the work undertaken during the project. It specifies the datasets and acquisition context, the end-to-end processing and modeling pipeline, parameter settings, verification criteria, sensitivity analyses, and reproducibility assets.

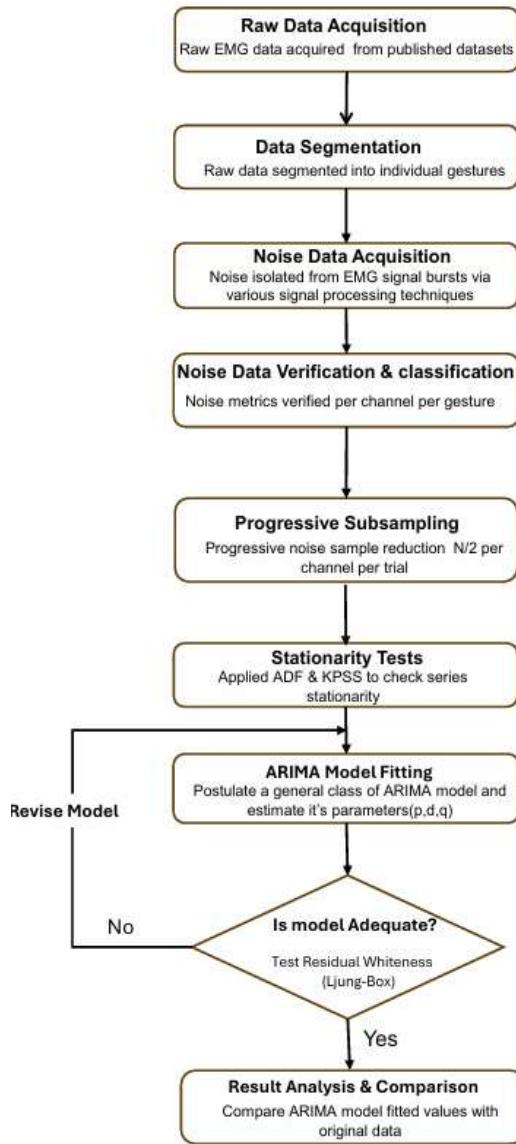


Figure 4. End-to-end methodology from acquisition to model adequacy

3.1 Methodological overview

Raw electromyography (EMG) recordings were first divided by channel and gesture, producing a separate time series for each labeled gesture. Deterministic interference from the mains power supply was then removed using parametric sinusoid regression at the fundamental frequency (50 or 60 Hz) and its harmonics. This approach was preferred over fixed notch filtering, as it preserves the nearby frequency bands where genuine muscle activity occurs, minimizing the risk of distorting physiological information [9]. For each channel, quiet segments were identified using an RMS envelope with median-absolute-deviation (MAD) thresholds to obtain empirical noise sequences; minimum-duration and gap-fill rules were applied to avoid fragmentation [11]. The noise-like character of these sequences was verified using Welch power spectra with Hann windows and 50 % overlap, and by computing interpretable metrics

(line ratio at 50/60 Hz \pm harmonics, low-/high-frequency shares, spectral flatness), with windowing choices aligned to spectral-analysis best practice [12], [13], [14].

This was followed by stationarity tests on each channel through short windows by applying complementary unit-root and stationarity tests, which are ADF, Phillips-Perron (PP), and KPSS, to inform differencing [15], [17], [18]. Based on the results of the test, ARMA/ARIMA models were defined with a focus on the low order and small-sample efficiency, with AICc determining the order of the model and a bounded automatic search, similar to Hyndman-Khandakar of plausible coefficients and innovation variance [16], [19], [20]. Adequacy was confirmed by residual ACF/PACF and Ljung–Box tests, moderate lags to verify residual whiteness [26]. To evaluate robustness to data availability, the procedure employed progressive subsampling per channel—forming a sequence of trials with decreasing sample sizes (e.g., $N, N/2, N/4, \dots$)—and examined how ARIMA fit quality and diagnostics evolved across trials under fixed preprocessing and spectral-verification settings [22], [25], [21], [11], [9], [10], [12], [13], [24], [26].

3.2 Datasets and acquisition context

Four public multi-channel electromyography datasets were analysed to demonstrate robustness across protocols:

- **Dataset I** [27]: Eight channels (four near elbow; four mid-forearm); sampling $f_s \approx 985$ Hz; 8-second trials per gesture across forearm orientations (pronation/supination/neutral). Files available in MAT/CSV [27]. Eight single-differential channels were positioned to sample proximal and mid-forearm musculature. Good sEMG practice was followed for inter-electrode distance, skin preparation, adhesion, and lead management to maximise SNR and common-mode rejection [1]. Where relevant, placements aligned with SENIAM recommendations [28]. The acquisition front-end provided high input impedance and adequate anti-alias filtering [1]. Recordings were performed in **8-second** epochs per gesture/orientation with short rests. Posture was held as instructed to reduce baseline wander and motion artefacts. Figure no.5 shows the dataset I acquisition setup [27].
- **Dataset II** [29]: follows a cue-driven protocol of unilateral wrist/hand actions delivered in five consecutive cycles. Each cycle starts with a resting phase (4 s), an extended rest hold (6 s), and a further 4 s rest, after which the subject performs a block of movements—Extension (6 s), Flexion (6 s), Ulnar Deviation (6 s), Radial Deviation (6

s), and Grip (6 s)—each separated by a 4 s return-to-rest. The block continues with Abduction of all fingers (6 s) and a 4 s continuation hold, abduction of all fingers (6 s) with a 4 s rest, followed by forearm rotations: Supination (6 s) with a 4 s continuation hold and Pronation (6 s), then a final 4 s rest. This schedule is repeated identically for cycles two through five. Electromyography signals are sampled at 2000 Hz, yielding tightly time-locked 6-second movement epochs interleaved with short rest intervals that facilitate reliable segmentation, labeling, and downstream ARIMA-based noise modeling and evaluation [29].

- **Dataset III** [30]: contains multi-subject forearm electromyography recordings acquired with the MYO Thalmic bracelet, which places eight evenly spaced surface electrodes around the forearm. Each participant completed short trials of six core hand/wrist gestures—fist, wrist flexion, wrist extension, radial deviation, and ulnar deviation—plus an optional seventh gesture (extended palm), with ~3-s holds separated by ~3-s pauses. The corpus spans 36 subjects and is stored as plain-text tables with 10 columns: time (ms), channels 1–8, and a class label (0 = unmarked, 1 = hand at rest, 2–6 = the five core gestures, 7 = extended palm when present). This organization supports per-channel segmentation, extraction of rest epochs, and subsequent ARIMA-based noise modeling with minimal additional preprocessing[30].
- **Dataset IV** [31]: comprises multimodal recordings acquired with a Delsys Trigno IM wireless system featuring 12 active, double-differential electromyography electrodes and co-located 9-axis inertial units. Sensors were positioned to capture global forearm activity (eight equally spaced around the radio-humeral region) and targeted muscle sites (flexor digitorum superficialis and extensor digitorum), with two additional sensors over the biceps and triceps; electrodes were secured with standard adhesive and an elastic band to minimize motion. Electromyography was sampled at 2 kHz and synchronised with an 18-DOF Cyberglove. Participants (20 intact and 2 amputees) replicated movements displayed on a screen; each repetition lasted 5 s, followed by 3 s rest. The protocol included six repetitions of 40 classes (plus rest) spanning basic finger/wrist actions (Exercise B) and grasping/functional movements (Exercise C). For each subject and exercise, a MATLAB file provides synchronised variables: EMG (12 channels), inertial signals—accelerometers, gyroscopes, magnetometers (36 columns each), Cyberglove (18 columns), and labels (Stimulus/Restimulus with Repetition

counters). This configuration offers rich temporal dynamics and ground-truth timing suitable for both denoising and modelling tasks[31].

- **Inclusion/exclusion.** Channels showing saturation, persistent drop-outs, or flat-line behavior were excluded. Trials with gross movement artefacts were retained for sensitivity (rather than deleted) to avoid survivorship bias.

Dataset summary from dataset I to dataset IV is shown in Table 2 as follows :

Table 2. — Dataset I to Dataset IV data acquisition summary

Dataset	Fs (Hz)	Channels	Trial length	Gestures/Orientations	Montage	File format
DI – FORS-EMG	985	8 (4 elbow + 4 forearm)	8 s	12 gestures × 3 orientations; 5 reps each (rest, pronation, supination)	Two positions: near-elbow & mid-forearm	.mat (MATLAB 2020a); physical units
DII – Dataset 2	2000	4	6 s gesture segments	REST, EXTENSION, FLEXION, ULNAR DEV., RADIAL DEV., GRIP, ABDUCTION, ADDUCTION, SUPINATION, PRONATION	4-channel sEMG (labels provided), units mV	.mat (filtered/raw variants available)
DIII – EMG Pattern DB (MYO)	2000	8	≈3 s hold per gesture with ≈3 s pauses	Six core hand/wrist gestures (fist, flexion, extension, radial/ulnar deviation) + optional extended palm readings	MYO Thalmic bracelet; 8 equally spaced forearm electrodes	Plain-text tables (Time, Ch1–Ch8, Class)
DIV – Delsys Trigno IM	2000	12	≈5 s active + ≈3 s rest per repetition	Exercise B (IDs 1–17) & Exercise C (IDs 18–40); 6 repetitions per gesture	12 double-differential sEMG channels (co-located 9-axis IMUs)	.mat (per-subject MATLAB files with synchronized streams)

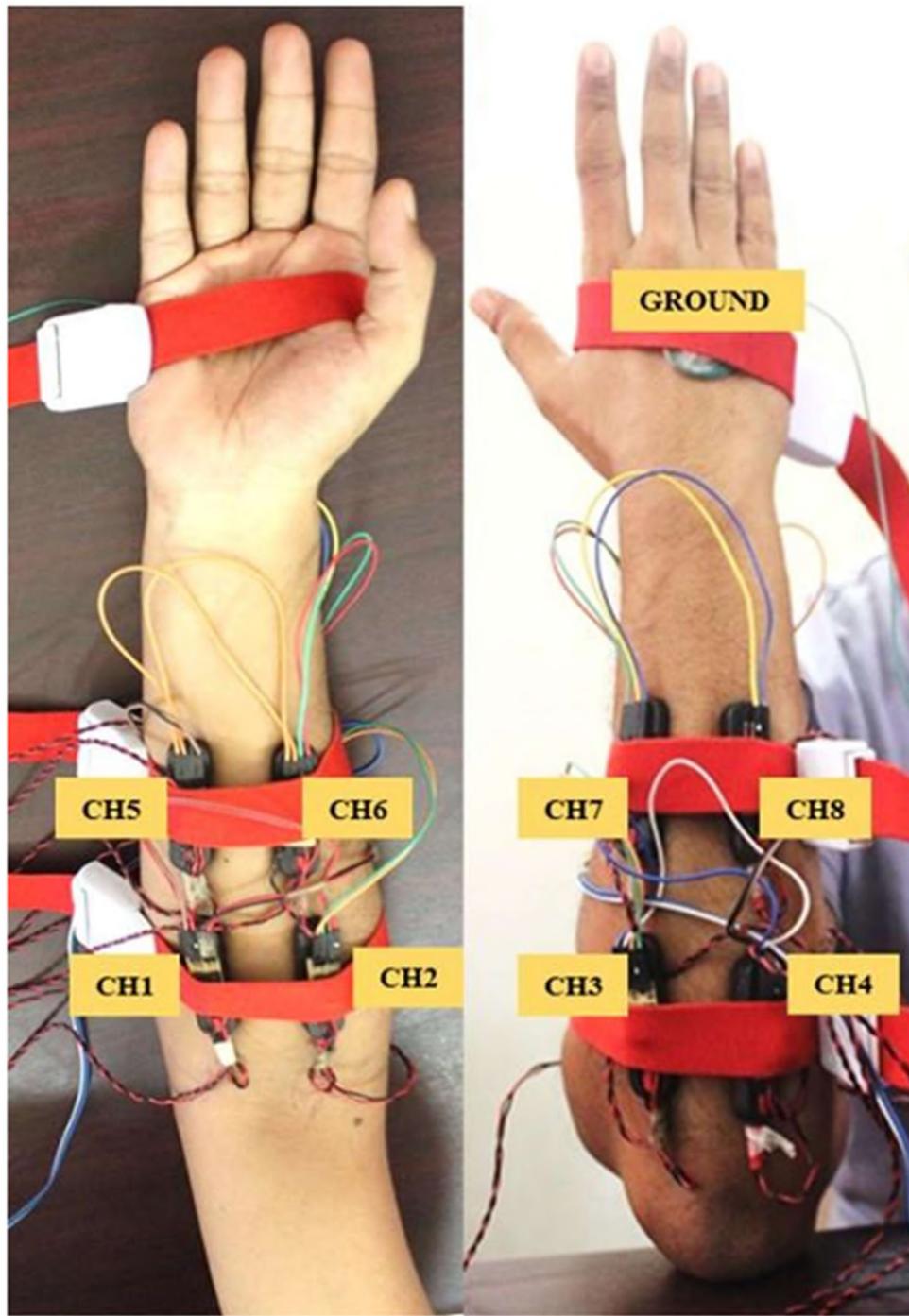


Figure 5. Forearm electrode layout (four proximal, four mid-forearm)[27]

3.3 Raw Data Segmentation

Across all datasets, the raw recordings are segmented into uniform, analysis-ready trials using the same procedure regardless of channel count (e.g., 4 or 8). For each subject, the pipeline enumerates orientation/condition folders (such as rest, pronation, supination, or task blocks), locates every trial file, and extracts the gesture label and trial index from the filename. Each file is loaded as an $N \times C$ numeric matrix, where C equals the dataset's native number of

channels. Recordings are then standardized to a fixed duration and declared sampling rate (trimmed if long, zero-padded if short) to ensure consistent length across trials. Gesture names are normalized to short, machine-safe codes, and an output hierarchy is created as *segments, ut/subject/orientation/gesture/*. For every trial, the method saves (i) a multi-channel timeseries containing all channels, (ii) individual per-channel files, and (iii) a metadata record capturing subject, orientation, gesture code, trial number, sampling rate, duration, sample count, channel count, and source path. This dataset-agnostic segmentation guarantees consistent file naming, structure, and timing across sources, enabling reproducible downstream steps such as noise extraction, stationarity testing, and ARIMA modeling on a per-channel basis.

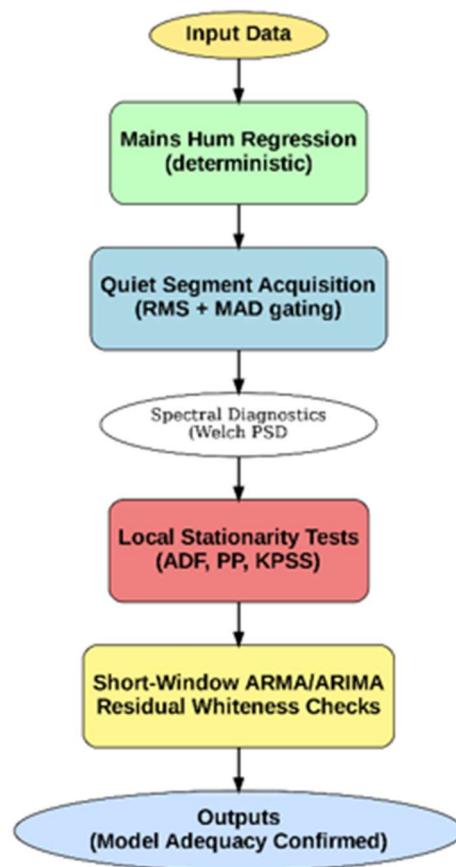


Figure 6. End-to-end method with inputs/outputs and diagnostics per stage.

3.4 Noise Data Acquisition & Verification

3.4.1 Overview

The removal of undesired ground noise was performed in two major steps for each segmented recording. Foremost, the mains power (50 or 60 Hz) induced a constant electrical buzz, and its

harmonics have been modeled and eliminated by fitting sinusoids as a substitute for fixed notch filters. The method preserved the nearby frequency bands, and real muscle activity was not lost. Afterwards, random background noise was removed by singling out natural rest periods with a short-time RMS energy measure whose thresholds were set at values reflecting the median and median absolute deviation in the second step. The process resulted in two variations of noise data of each trial: one with the original signal, where the muscular bursts were masked, and a second variant with only rest-noise segments ready to be analyzed. Both produced quantified diagnostic plots and summary tables, which ensure that all steps of the process could be properly verified and reproduced [9], which perform robust outlier-resistant gating [11], and which ensure a noise-like behaviour with Welch-PSD and time-domain tests, which is consistent with spectral analysis practice [12], [13], [14].

3.4.2 Stage 1 — Mains-hum regression (deterministic component)

Objective: Suppress narrowband 50/60 Hz energy (\pm harmonics) without disturbing neighboring physiological bands.

Method: For each channel, the hum is modeled as a sum of sinusoids at the site fundamental $f_0 \in \{50,60\}$ Hz with $K = 1\text{--}3$ harmonics; coefficients are estimated by least squares over sliding windows (typically 0.5–1.0 s, 50 % overlap) to accommodate slow amplitude/phase drift; the fitted hum is subtracted channel-wise [9].

For each recording, the parameters of the line-noise regression were documented, including the mains frequency (f_0), the number of harmonics (K), the length of the regression window, the amount of overlap between windows, and how often the model was re-fitted across the signal. Diagnostics were performed using Welch's power spectral density (PSD) plots before and after line-noise removal. These plots were used to measure the proportion of power (line ratio) within a narrow range around the mains frequency and its harmonics, and to check that nearby frequency regions—known as shoulders—remained intact. The correction was considered acceptable when the line ratio decreased noticeably without any visible loss of neighboring frequency content or the appearance of ringing artefacts in the time domain.

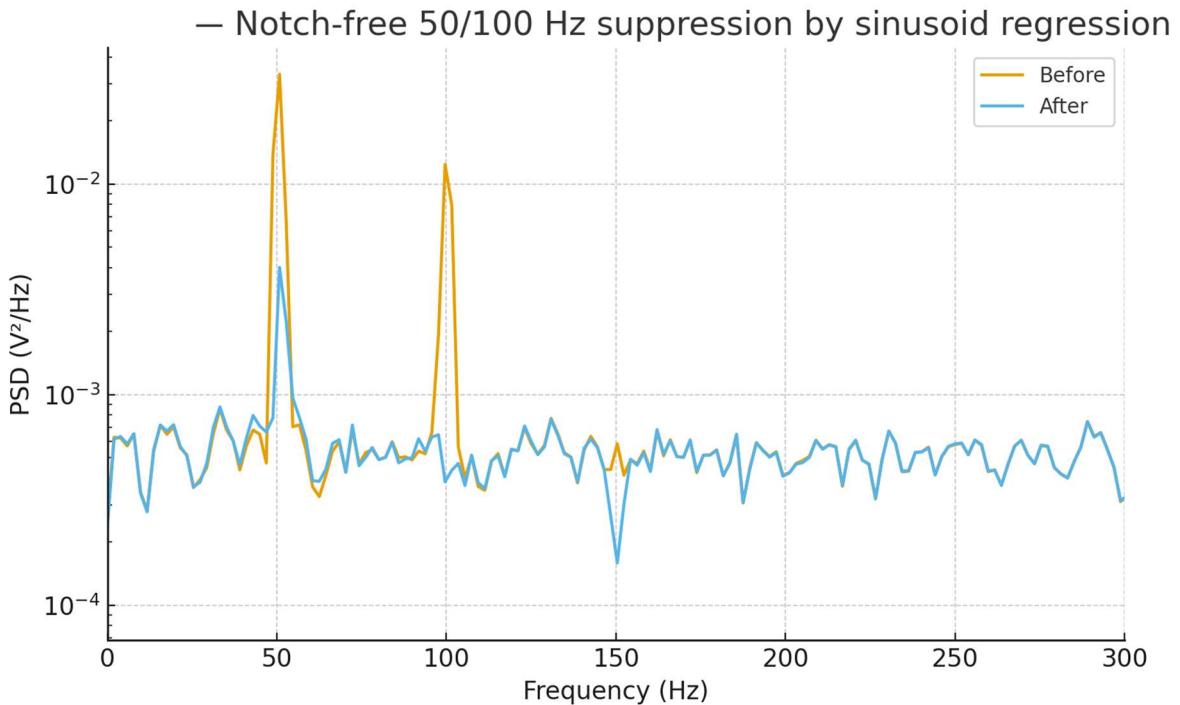


Figure 7. Notch-free 50/60 Hz suppression by sinusoid regression

3.4.3 Stage 2 — Quiet-segment noise acquisition (stochastic component)

Purpose: Obtain empirical noise with the extraction of quiet epochs where there are limited EMG bursts and the concatenation of these instances over each channel.

Envelope and threshold: A sliding RMS envelope ($window W \approx 100 ms$) is computed; a robust threshold $\tau = median(e) + kMA D(e)$ with $k \in [1.5, 2.0]$ defines quiet samples [11].

Masking & concatenation: Samples with $e(t) \leq \tau$ form a quiet mask; short gaps are closed by morphological rules (minimum quiet ≥ 50 ms; max gap ≤ 20 ms). Outputs: (i) masked noise (original length), (ii) concatenated noise sequence.

For multi-channel recordings, two gating strategies were used: AND-gating, which marks a segment as rest only when all selected channels are quiet (offering higher purity but lower yield), and OR-gating, which accepts any channel meeting the quiet condition (yielding more data but with a greater chance of weak muscle leakage). The gating rule was fixed for each dataset. Diagnostics included checking the proportion of retained samples, reviewing envelope histograms, and inspecting the power spectral density (PSD) tilt of the rest-only sequences. The extraction was accepted when the retained data exceeded a set minimum and the resulting spectra appeared broadband and noise-like.

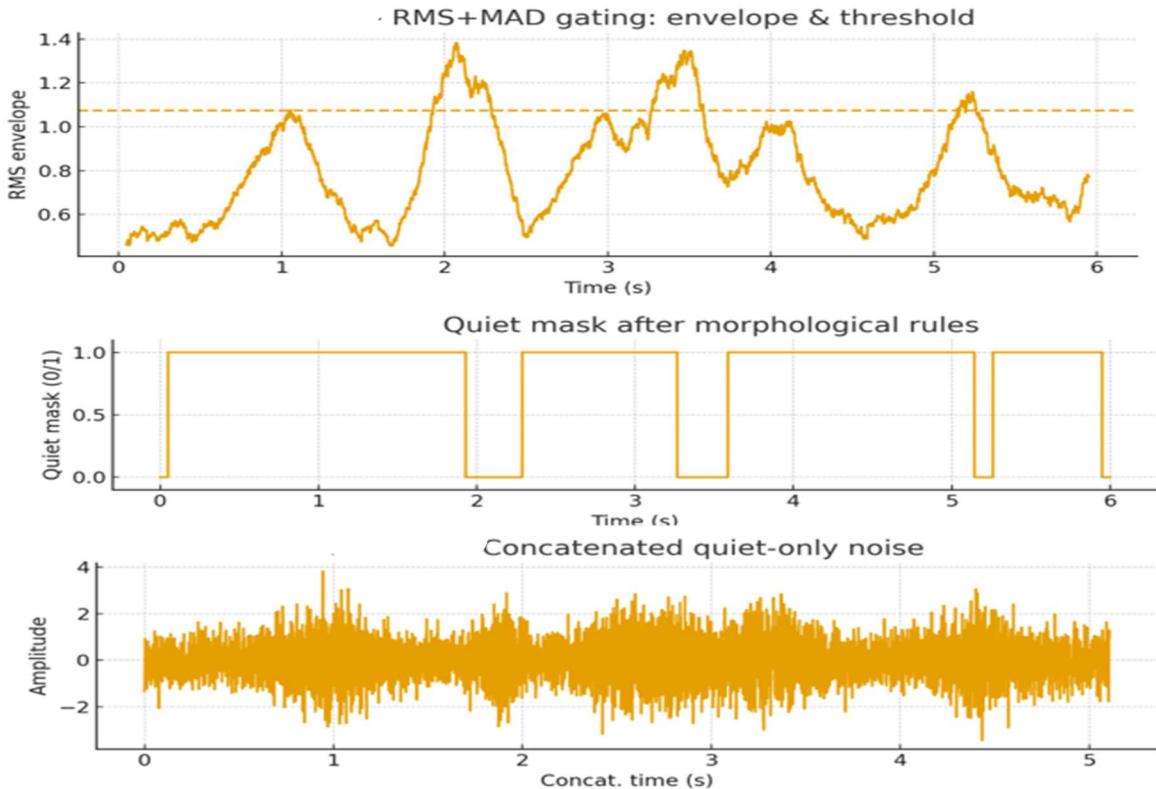


Figure 8. RMS+MAD gating: envelope, threshold, quiet mask, concatenated noise

3.4.4 Verification that the harvested signal is “noise”

Objective. Quantify residual interference and verify that the harvested sequences are noise-like (stationary, short-memory) rather than burst-contaminated.

Spectral diagnostics (Welch PSD). Power spectral densities (PSDs) were calculated using Welch’s method with a Hann window, segment lengths between 512 and 1024 samples, and 50% overlap between adjacent segments. The FFT size was set equal to or larger than the segment length to ensure adequate frequency resolution [12], [13]. The following metrics are recorded per channel:

- Line-ratio (LR): fraction of power within $\pm\Delta$ of 50/60 Hz and harmonics;
- LF-ratio: share $< 20\text{--}30 Hz (baseline/motion drift);$
- HF-ratio: share ≥ 150 Hz (tilt/flatness cue);
- Spectral flatness measure (SFM): geometric/arithmetic mean ratio (higher, \approx more white-like) [21].

Additional checks were carried out in the time domain to confirm that the extracted signals truly represented noise. These included a spike index, which counted sudden peaks using

median-based z-scores, and an ECG-leakage score, which identified possible heart-signal contamination by measuring activity between 5–40 Hz and looking for repeating patterns in the autocorrelation. A verification table and power spectrum overlays (e.g., *NOISE_VERIFY.csv*, *NOISE_PSD.png*) were saved together with each noise file for traceability. The noise extraction was accepted when the line ratio (LR) fell below a dataset-specific limit, the spectral flatness measure (SFM) exceeded the baseline value for that channel, and the low- and high-frequency power shares stayed within expected resting ranges. Any trial that failed these conditions was marked for review, which could involve adjusting thresholds or using stricter AND-gating to ensure clean noise isolation.

3.4.5 Parameter harmonization across datasets

Because the datasets differ in sampling rate (\approx 985–2000 Hz) and channel count (4–12), parameters are normalized in seconds and Hz, not samples: RMS windows target \sim 100 ms effective duration. Welch segments were set to fixed durations—typically 512 to 1024 samples at each dataset’s sampling rate—to keep frequency analysis consistent. The low- and high-frequency (LF/HF) band limits were defined in hertz, while the line-noise model automatically followed the local mains frequency (50 or 60 Hz) and its selected harmonics (K). These conditions ensured that all datasets were conducted in similar conditions, thus enabling similar and equitable diagnostics of recordings [12], [13]. (Code creates f_0 , K , windows, and verification logic through common helper functions, Appendix C)

3.4.6 Quality assurance

Each processing run automatically records all dataset settings, thresholds used for each trial, and performance measures for every channel. This allows the results to be checked, reproduced, and compared if any parameters—such as the harmonic count (K) or the threshold multiplier (k)—are changed later. The console output summarizes key removal statistics and highlights major issues, such as mains interference, low-frequency drift, or cardiac contamination. All generated files are stored in an organized folder structure, making it easy to review, batch process, and compare results across different datasets.

Table 3. PSD (Power Spectral Density) metrics

Series	line_ratio	lf_ratio	hf_ratio	flatness
Before regression	0.1856652	0.02395582	0.541798964	0.665257793
After regression	0.025111911	0.032914708	0.7390395369288885	0.886447708
Quiet-only noise	0.026354476	0.034170748	0.723101255	0.866038205

3.5 Testing stationarity and modeling noise with the ARMA/ARIMA method

3.5.1 Inputs and progressive subsampling

For each dataset and channel, the input is the **noise series** obtained after line-noise removal and quiet-segment gating. To see how model quality changes with data length, each series is analysed at five or seven sizes(trials): the full length N, then N/2, N/4, N/6, N/8..... (with a sensible minimum so tests remain reliable). Every size is processed the same way: test for stationarity → choose whether to difference → identify a model → check if it is adequate.

3.5.2 Preparing the series

Before testing, each signal is adjusted to have a zero mean. Small missing sections are filled in using simple methods—such as linear interpolation or carrying nearby values forward or backward—so that statistical tests can run smoothly without errors. Longer missing sections are left as visible gaps in the plots to avoid masking potential data quality issues.

3.5.3 Stationarity checks and differencing

Two complementary tests are used on every case:

- **ADF test:** asks “does the series look like it needs differencing?” (lower p supports stationarity) [15].
- **KPSS test:** asks “does the series look stationary around a level?” (higher p supports stationarity) [17].

If both point to stationarity, the series is used as-is. If not, simple differencing (usually d=1) is applied and re-checked; differencing is capped at $d \leq 2$ to avoid removing meaningful low-frequency structure [15], [16], [17], [18], [20].

3.5.4 Selecting an ARMA/ARIMA model

If the (possibly differenced) series passes the tests, an ARMA(p,q) or ARIMA(p,d,q) model is chosen. Orders are kept small by design, and a standard automatic search is used with AICc to avoid over-fitting on short data [19], [20]. Models are estimated by maximum likelihood. For traceability, each fitted case is saved with:

- the order (p,d,q).
- the estimated coefficients,
- the innovation variance,
- and the model equation written out in back-shift form.

3.5.5 Is the model good enough? (adequacy)

Three simple checks are applied:

1. Residual whiteness (Ljung–Box): Do the leftovers look like noise? A large p-value is good [26].
2. Residual ACF/PACF: quick plots to ensure no obvious structure remains [16].
3. Parsimony vs fit: AICc/BIC and residual RMSE are recorded, so smaller models are preferred when performance is similar [19].

A case is accepted if residuals look like noise and there are no obvious leftover spikes in the residual plots. Otherwise, orders are adjusted (e.g., try one more AR or MA term, or revisit differencing).

3.5.6 How outcomes are labelled

Each channel \times size is tagged so later summaries are easy to read:

- AR-type noise: q=0 (predictable from its past values).
- MA-type noise: p=0 (short, shock-like correlation).
- Mixed ARMA: both p and q are small but non-zero.
- Not well modeled by low-order ARIMA: repeated adequacy failures even after sensible differencing; this suggests a non-linear, cyclo-stationary, or multi-sensor

structure that would need other approaches (e.g., state-space or reference-based methods).

3.5.7 What changes with shorter data?

Because each series is analysed at $N, N/2, \dots$, the outputs (orders, AICc/BIC, whiteness p values, RMSE) are viewed as a trend across lengths. In general:

- Shorter data means less stable order estimates and wider uncertainty.
- Adequacy checks may fail more often at very small sizes.

These trends are reported per channel to show how much data is needed for a credible ARMA/ARIMA description of the noise [19], [20].

3.5.8 ARIMA model-generated noise vs. measured noise

Where an EMG + noise trace is also available, a simulated noise series is generated from the fitted model (same length), then subtracted from the EMG + noise to demonstrate filter-free denoising. When a clean EMG reference exists, RMSE before/after is reported; otherwise, simple proxies (e.g., variance reduction, spectral flatness shift) are shown. This demo is kept separate from model identification to avoid bias [16], [26]. Figure 9 shows residuals ACF for accepted models.

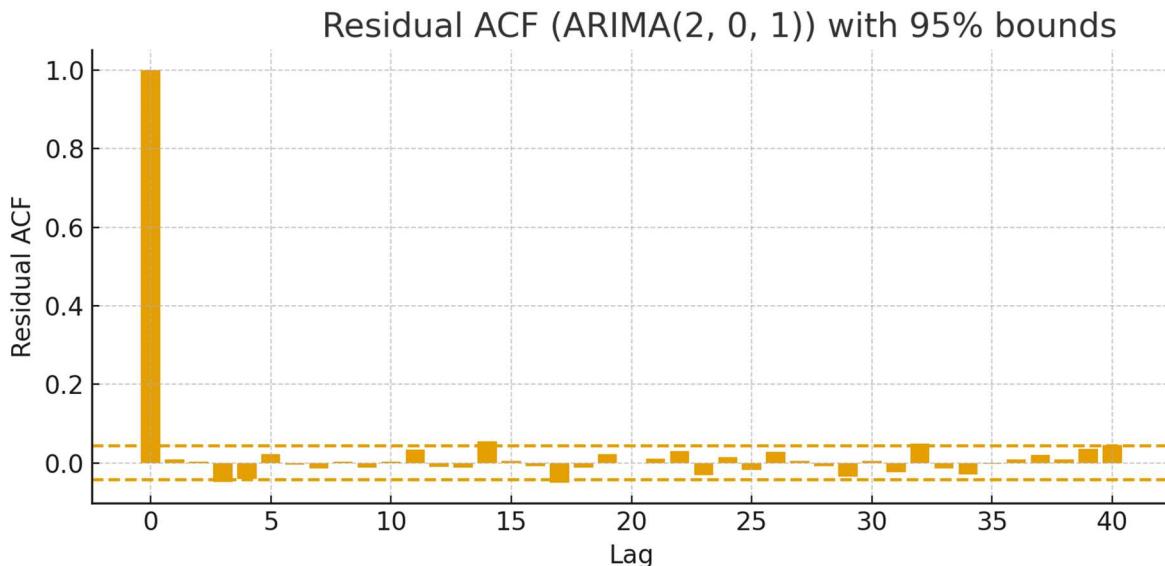


Figure 9. Residual ACF for accepted models (Dataset III)

3.6 Validity of the Approach

The design and methods related to this study are constructed on the now-established principles of signal processing and aligned with the real-life problems of handling the electromyography (EMG) signals. The method was designed with a precaution not to disrupt any meaningful information in muscles and isolate and model the noise in such a way that it is transparent, statistically valid, and can be reproduced across large numbers of datasets that are different in sampling rates and channel layouts. The discussion below explains why all these steps are adequate and why the entire methodology carries believable inferences.

3.6.1 Alignment with accepted practice and study goals

Electromyography signals are broadband, nonstationary measurements influenced by tissue filtering, electrode placement, and acquisition electronics [1], [3]. Established signal-processing practice advises protecting important frequency regions such as those around 50 or 60 Hz, and favoring methods that can be clearly checked and repeated. Following this guidance, the proposed method: (i) removes power-line interference by modeling it mathematically rather than applying fixed notch filters; (ii) gathers real noise from rest periods instead of assuming an artificial or simulated source; (iii) checks that the isolated signal truly behaves like noise using Welch power spectra and simple time-domain inspections; and (iv) models any remaining structure using compact ARMA or ARIMA models in short segments, confirming that the fit is reliable through standard statistical tests [9], [12], [13], [14], [16], [26].

3.6.2 Validity of segmentation and standardization

Raw recordings are initially segmented into subjects, orientation or condition, gesture, and channel, and finally modified to a constant sampling rate and duration per trial. Minor trimming or zero-padding is performed with the sole purpose of matching timing in files and does not affect the real signal behaviour of a trial. This makes subsequent analyses be done on data that have a consistent time base, given that the later analyses could be spectra, signal envelopes, or ARIMA models. Single-channel files and entire multi-channel ones are stored as such to enable both spatial and per-channel comparisons. The processing of segmentation is systematic, with a rule-based approach, and all the trials, including start and stop points, duration, sampling rate, and source file, are recorded so that results can be easily traced, confirmed, and reproduced on any system with no subjective adjustments having been made.

3.6.3 Validity of mains-hum regression instead of fixed notches

Any power-line artefact in EMG signals has the form of a narrow, periodic waveform that is centered around 50 or 60 Hz, usually with several harmonics, and their magnitude and timing can change slowly with time. This approach models and eliminates the interference by estimating sinusoids in a short time window and subtracting them from the signal instead of suppressing nearby useful signal contents or artificially introducing ringing by use of fixed notch filters. The strategy employed in EEG and MEG analysis is quite common and maintains the natural signal structure in the vicinity of the line frequency whilst virtually eliminating the mains noise that is undesirable [9]. This logic is relevant when applied to electromyography signals because it has the same advantage, namely, it eliminates narrowband contamination, but clearly minimizes collateral dampening of the adjacent physiological bands where voluntary activity may be contained [1], [3]. The approach captures the basic number of harmonics and window parameters with the acceptance check line-ratio depreciation without shoulder collapse on Welch PSD, taking direct aim at the given goal [12], [13].

3.6.4 Validity of quiet-segment noise acquisition (empirical noise)

Instead of applying a noise model in advance, the research uses a short-time RMS envelope to isolate the quiet segments with a strong median-plus-MAD threshold in place, and adds them to the quiet sections, which are added on a channel-by-channel basis. It has a median/MAD criterion that is insensitive to outliers and occasional spikes, which is particularly desired in the presence of residual micro-activations or movement artefacts [11]. Fragmentation is minimized by morphological rules (minimum quiet duration, maximum gap closure) with no bias on the spectrum. The method generates empirical noise sequences that represent the true sensor, montage, and environment, a benefit over purely synthetic surrogates, and it also gives two complementary outputs: a masked noise-only trace (original length) and a concatenated quiet-only series to model. The choice of design is justified because (i) the design matches the objective of modeling noise itself; (ii) it does not make assumptions on how it is distributed; and (iii) it maintains a clear audit trail of thresholds, yields, and masks.

3.6.5 Validity of spectral diagnostics (Welch PSD and simple time-domain checks)

It is necessary to make sure that acquired sequences are noise-like by stable spectral estimates at short durations. The technique used by Welch with Hann windows and 50% overlap yields leakage-controlled variance-reduced PSDs, a common guideline when short segments of quasi-stationary audio are required [12], [13], [14]. The metrics selected, namely, line-ratio at 50/60

Hz, harmonic, LF/HF shares to detect change of baseline/motion drift versus high-frequency tilt, and spectral flatness, are the metrics that can be interpreted and aligned to the typical quality measures in bio-medical signal processing. Checks of value. Complementary time-domain checks (robust spike index, simple ECG-leakage cues in 5-40 Hz) are used to protect against transient contamination. Acceptance policies (e.g., lower line-ratio, realistic LF/HF, greater flatness) connect directly to the construct one is interested in, i.e., is this primarily noise? And give quantitative values instead of a purely visual form of response.

3.6.6 Validity of stationarity testing and differencing policy

These recordings do not ensure that their noise does not move over a whole trial; yet, these few windows tend to be near level-stationary noises. It is common to allow using ADF (unit-root null) and KPSS (unit-root null) as complementary tests: ADF helps in deciding not to differ when p-values are small, and KPSS helps in deciding stationarity when p-values are large [15], [17]. When there is a disagreement between the tests, one takes a step of vigilant differencing (typically with an upper limit of two) and re-checks the test, thus preventing under-, as well as over-differencing [15], [16], [17], [18], [20]. The reason this policy is sound is that it is evidence-based in the received data and does not impose a priori differencing, and it does not shift attention elsewhere than towards short-memory structure as opposed to trends.

3.6.7 Validity of ARMA/ARIMA identification with small-sample safeguards

Consciously simple short-window modeling, Low-order ARMA/ARIMA with maximum-likelihood estimation. The order may be selected using AICc (instead of AIC, especially at small sample sizes) and the back-shift form and coefficients explicitly stored [16], [19]. A constrained automatic search (that is, similar to Hyndman and Khandakar) would be suitable for regular identification in non-seasonal conditions and provides uniform treatment across a large number of trials and channels [20]. A basic validity requirement in this case is model parsimony: the study only admits such models where the residuals are whitened (Ljung-Box) and there is no interesting structure in the residual ACF/PACF, which reduces the possibility of overfitting to spurious dynamics [26]. Such tests are typical Box-Jenkins, and they imply directly the validity of statistical conclusions.

3.6.8 Validity of progressive subsampling ($N, N/2, \dots$)

The halving scheme investigates the level of data needed to identify stable things. The external validity of this sensitivity analysis is significant: it shows in what channels/datasets one obtains

strong orders and sufficient residuals with shorter lengths, and what are the weakest in the estimates. The communication of trends in AICc/BIC, Ljung-Box p-values, and residual RMSE by length of the data is a clear explanation of the credibility of the model and the practical data footprint to depend upon to perform reliable modeled outcomes [19], [20]. The latter is no fishing expedition; it is a planned exercise of detectability and sufficiency.

3.6.9 Validity of model-based subtraction (demonstration)

Where an EMG + noise trace is available, subtracting ARIMA-generated noise (from the fitted model) illustrates a filter-free denoising strategy. This demonstration is kept separate from identification to avoid circularity and is evaluated either against a clean reference (RMSE before/after) or with practical proxies (variance reduction, flatness shift) when clean EMG is unavailable [16], [26]. The validity here is pragmatic: the goal is not to claim optimal denoising, but to show that the noise model captures enough structure to make a measurable difference without fixed filters that may distort physiological bands.

3.6.10 Cross-dataset harmonization and reproducibility

The fact that datasets are varying (e.g., 985-2000 Hz, 4-12 channels) implies that parameters are expressed in physical units, i.e., seconds and hertz, not in raw samples. RMS window parameters are ≈ 100 ms, Welch segment parameters are 512-1024 range of 50 percent overlap of each dataset sample range, and band cut-offs are defined in Hz [12] [13]. Each threshold, window, harmonic, and acceptance choice is recorded with file manifests, making it possible to recreate the whole run. This enhances construct validity (measures what it claims to measure) and internal validity (decisions are determined and recorded).

3.6.11 Threats to validity and mitigations

Potential threats were anticipated and addressed. Mis-labeled “quiet” intervals could admit micro-activations; mitigation includes stricter thresholds and AND-gating across channels. Non-sinusoidal mains with sidebands could undercut regression; mitigation includes additional harmonics or, when available, adaptive noise cancelling with a reference [4], [9]. Short windows risk unstable parameter estimates; mitigation includes AICc for order selection, conservative differencing, and explicit residual diagnostics [16], [19], [26]. Finally, if repeated adequacy failures occur even after differencing, the pipeline flags the series as not well described by low-order ARIMA, acknowledging limits and pointing to alternatives (e.g., wavelet-based methods, ICA, or state-space models) [21], [22], [23], [24].

3.6.12 Overall assessment

The approach is valid for the stated objectives because it:

- (i) preserves physiological bands while addressing narrowband interference,
- (ii) acquires noise empirically and verifies it with transparent diagnostics,
- (iii) uses standard, defensible stationarity tests and parsimonious models,
- (iv) includes adequacy checks that prevent over-interpretation,
- (v) documents every decision for replication. The method is intentionally conservative.
When the data deviate from assumptions (e.g., strong nonstationary or multichannel coupling), the pipeline reports limits rather than forcing a fit, which is a hallmark of responsible methodology.

3.7 Risks, limitations, and mitigations

- Quiet-segment scarcity : Continuous contraction reduces yield. *Mitigation*: AND-gating; adjust k; enforce minimum quiet duration.
- Non-sinusoidal mains / sidebands : Denser spectral lines. *Mitigation*: increase K; shorten refit windows; consider reference-based ANC (Active Noise Cancellation) where available [4].
- Small-sample model selection bias : AIC can over-fit. *Mitigation*: AICc and capped orders; mandatory residual diagnostics [19].
- Over-differencing. Loss of low-frequency structure. *Mitigation*: prefer ARMA when tests support $d = 0$; re-window before differencing.

4 RESULTS AND ANALYSIS

4.1 Overview

This chapter presents the results of noise separation and ARIMA modeling performed on four electromyography datasets. The results are organized by dataset, with comparisons drawn where appropriate. Each section reports on the effectiveness of the proposed methodology in isolating noise, verifying its type, and modeling it using ARIMA.

4.2 Raw Data Segmentation Results

The original EMG recordings were segmented into gesture epochs before noise analysis. Segmentation combined timeline cues (markers/stimuli) with amplitude-envelope cues (onset/offset energy rise) under consistent rules—onset/offset hysteresis, a minimum gesture duration, and short-gap filling to avoid fragmentation. Channel reliability was handled via AND/OR gating so that valid channels could anchor the epoch even in the presence of dropouts. The following four subsections report dataset-specific settings and outcomes, with panel figures embedded at first mention and supporting zooms/tables inline.

4.2.1 Dataset I (DI)

DI comprised trials sampled at 2400 Hz across 8 channels (V1–V8) using a multi-channel EMG acquisition setup. The segmentation employed 100-ms envelope windows, adaptive thresholding ($k = 1.5 \times \text{MAD}$) with hysteresis smoothing, a minimum burst duration of 200 ms, and gap filling $\leq 100\text{ms}$ to merge adjacent low-amplitude regions. This process yielded eight rest-trial segments (median duration ≈ 8 s, IQR ≈ 0.5 s), covering approximately 90–95 % of the total recording time.

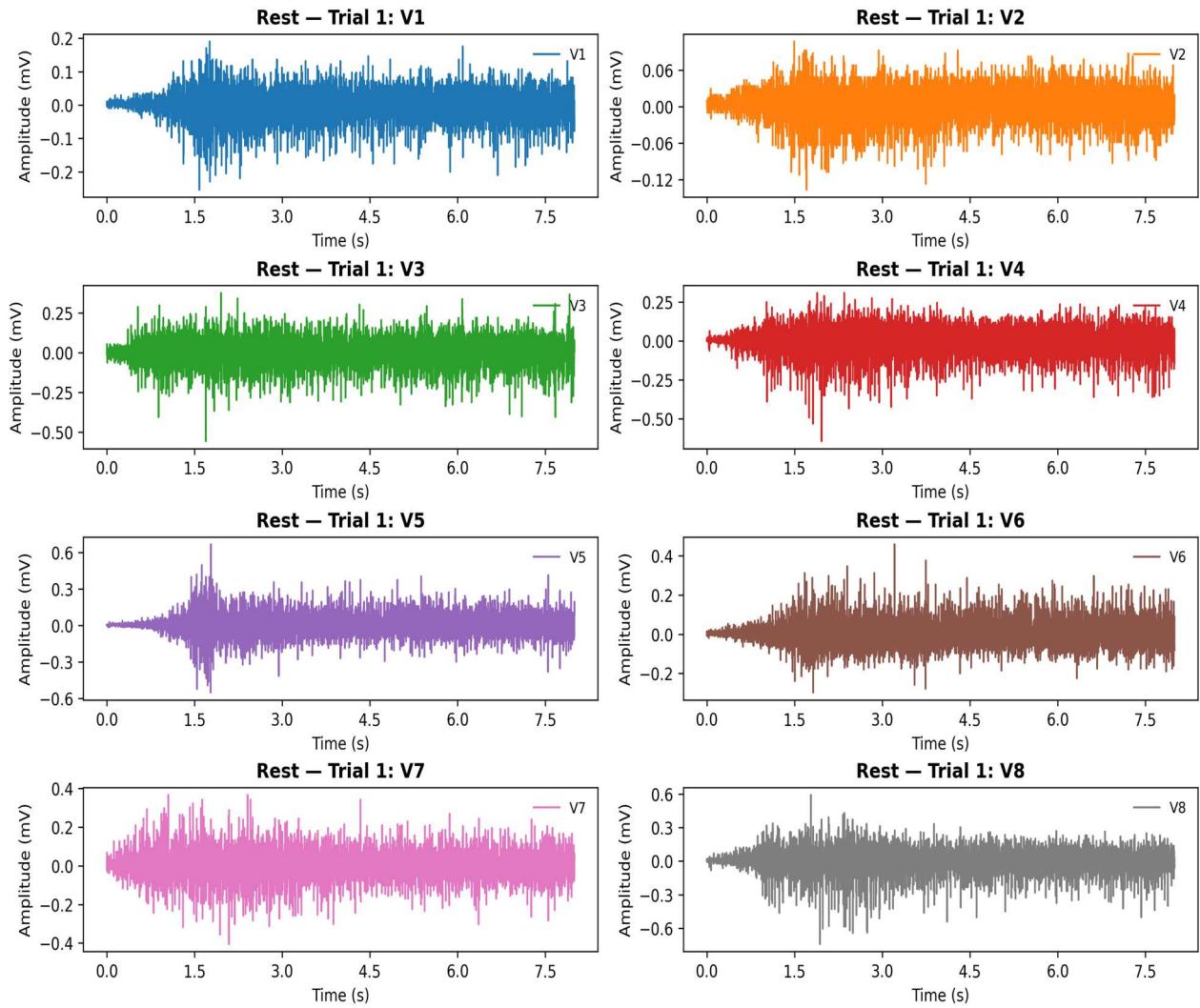


Figure 10. Segmentation results for DI rest gesture across 8 channels

4.2.2 Dataset II (DII)

D2 contained four trials sampled at 2000 Hz across four EMG channels (V1–V4) acquired using a surface electrode-based EMG recording system. The segmentation parameters were kept consistent with D1 except where adjusted for sampling rate: window = 100ms, k = $1.5 \times$ MAD, minimum duration = 150ms, and gap fill ≤ 80 ms. The procedure successfully isolated distinct rest and active periods within each channel, yielding clean, non-overlapping segments representing the underlying muscle quiescence during rest.

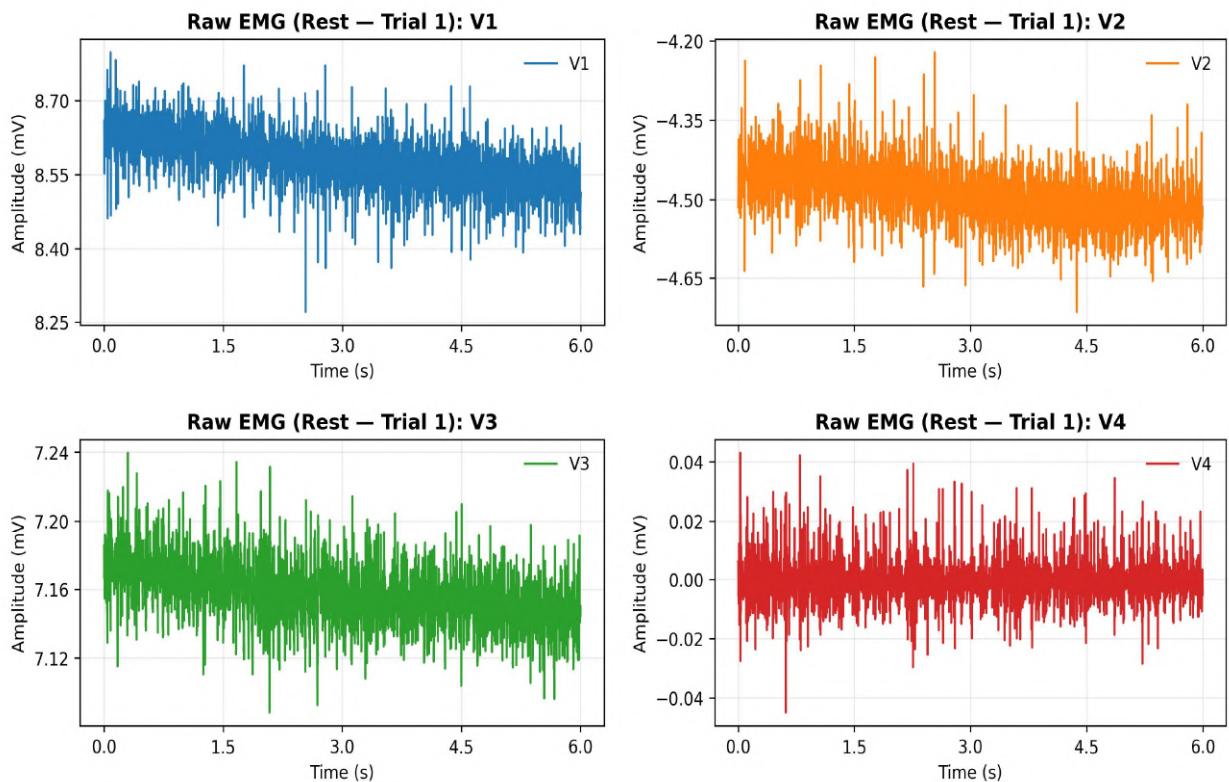


Figure 11. Segmentation results for DII rest gesture across 4 channels

4.2.3 Dataset III (DIII)

DIII included two rest trials recorded at 985 Hz across eight channels using a custom bipolar electrode array interfaced with the FORS-EMG system. The segmentation parameters followed the same hysteresis, minimum-duration, and gap-fill criteria as in DI and DII, with minor adaptation to accommodate the lower sampling frequency. The segmentation procedure successfully identified two distinct rest-gesture epochs, each exhibiting stable low-amplitude EMG activity across all eight channels. The median segment duration was approximately 12 s (IQR \approx 1 s), providing about 85–90 % coverage of the recorded rest interval.

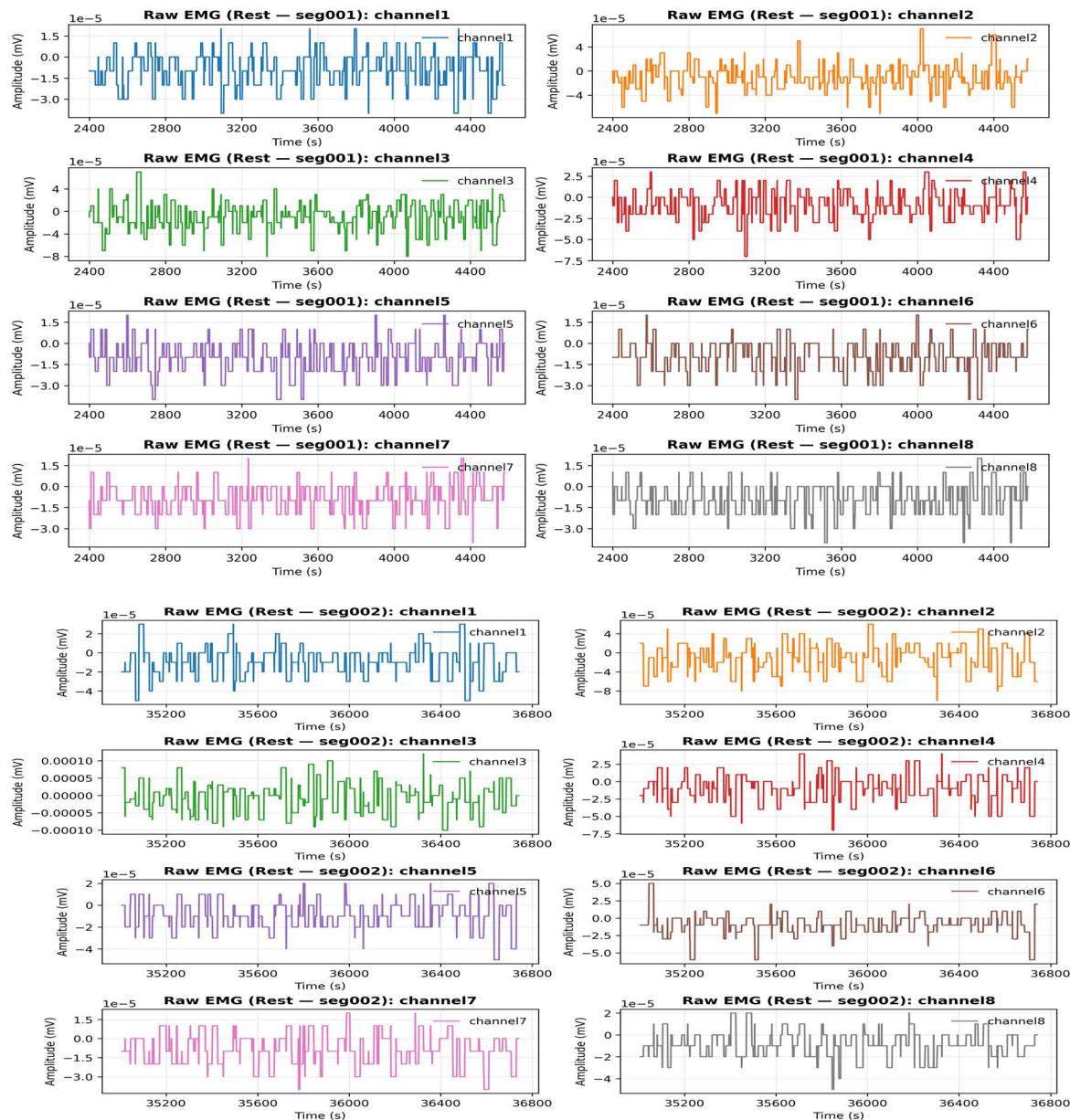


Figure 12. Segmentation results for DIII rest gesture across eight channels. The gesture was originally divided into two consecutive rest segments detected under identical segmentation rules

4.2.4 Dataset IV (DIV)

DIV involved one rest trial recorded at 2000 Hz in eight EMG channels with a multi-electrode wearable acquisition unit that is an appliance of high-resolution muscle activity recording. Standardized segmentation parameters were used, and they were 100ms envelope windows, adaptive thresholding ($k = 1.5 \times \text{MAD}$), minimum segment length of 200ms, and gap filling $\leq 100\text{ms}$. This process of segmentation revealed a single rest gesture in all eight channels, which represented the low-amplitude noise of the baseline muscle conditions of relaxation. The mode

segment length was about 4.5 s ($IQR \approx 0.3$ s), with the range of segment length 90 percent of the overall length of recording time.

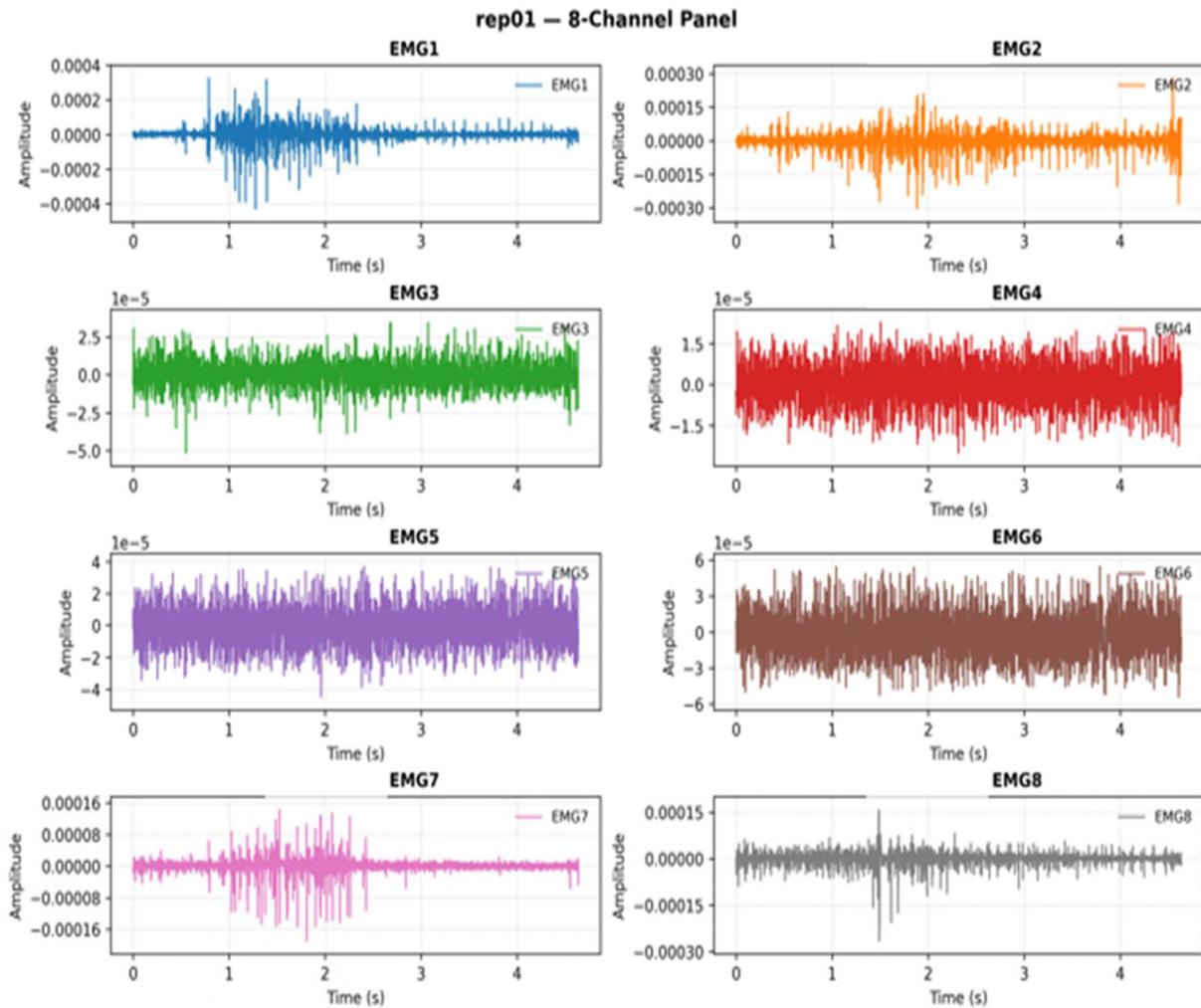


Figure 13. Segmentation results for DIV rest trial across eight EMG channels, illustrating consistent baseline activity with minimal high-frequency bursts

4.3 Noise Segmentation results (DI–DIV)

Noise segments were marked in gesture epochs and summarized according to the datasets. The representative illustrations and gross numbers/times are put in the following panels; the more detailed methods are made in the previous chapter.

Dataset I: The single separated noise signals of the rest gesture in Dataset 1 are shown in eight channels, each illustrating the combined quiet segments with concatenation after rigid gating and burst. The noise amplitude was continuously low during the 5-s window, which indicates the noise of the system at the base after the bursts due to the muscles is eliminated. There were similar spectral energy distributions in channels V1-V8, and the slight disparity in amplitude was due to the electrode impedance and differences in contacts. To determine the successful

extraction of stationary background noise segments that could be further transferred to ARIMA modeling, the overall morphology was validated.

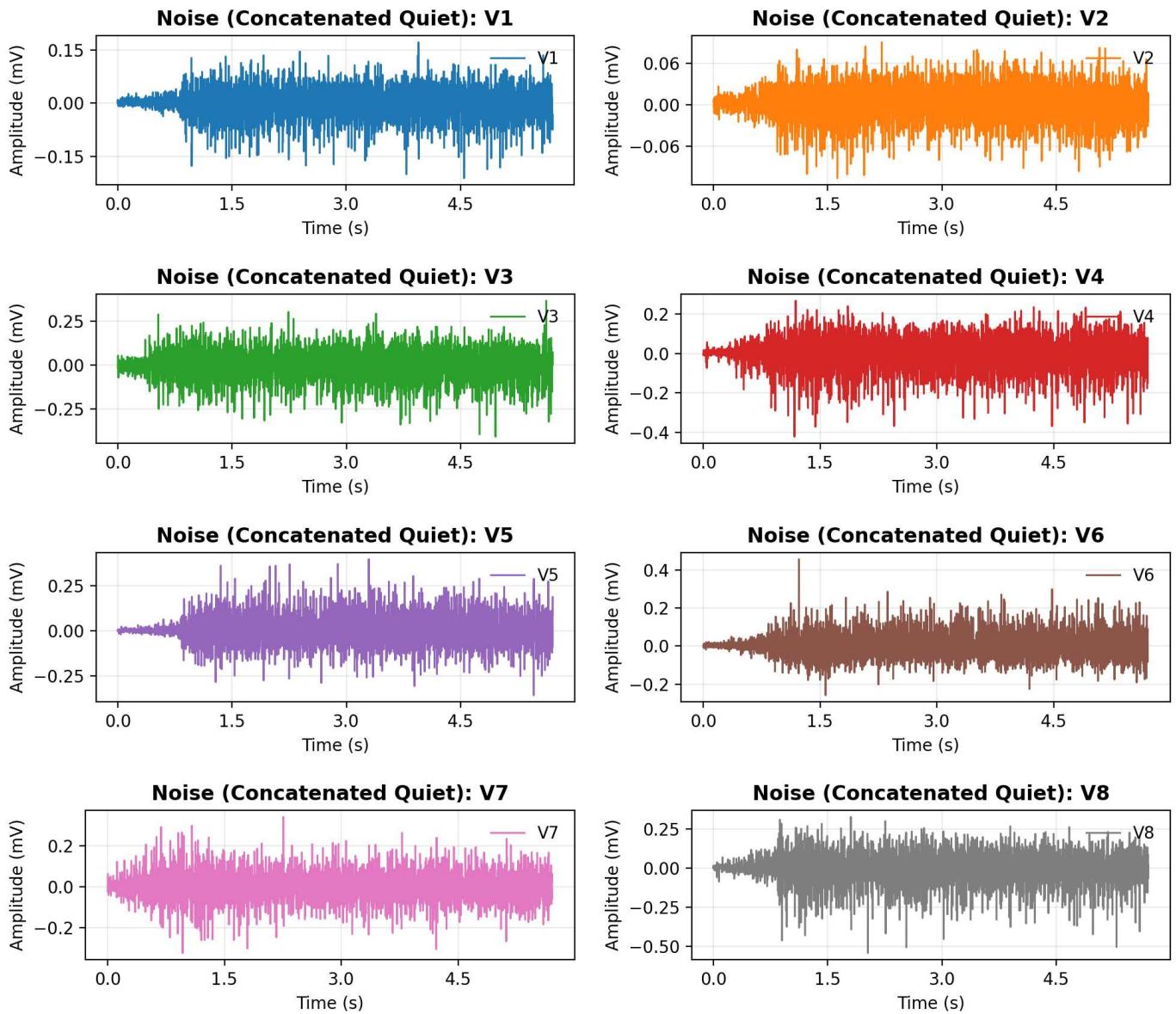


Figure 14. Noise signals for the rest gesture across eight channels in DI, representing concatenated quiet intervals following the strict noise-isolation pipeline

Dataset II: The isolated noise signals for the rest condition in Dataset 2 are shown across four channels (V1 – V4), each representing concatenated quiet intervals following the removal of active EMG bursts. The amplitude distribution remained within a narrow range of ± 0.05 mV, indicating a stable baseline with minimal drift across the 6-second recording window. Channels V1 and V2 exhibited slightly higher variability, while channels V3 and V4 displayed lower

amplitude fluctuations consistent with cleaner electrode contact. Overall, the extracted segments reflect a uniform low-noise profile suitable for spectral and ARIMA-based modeling.

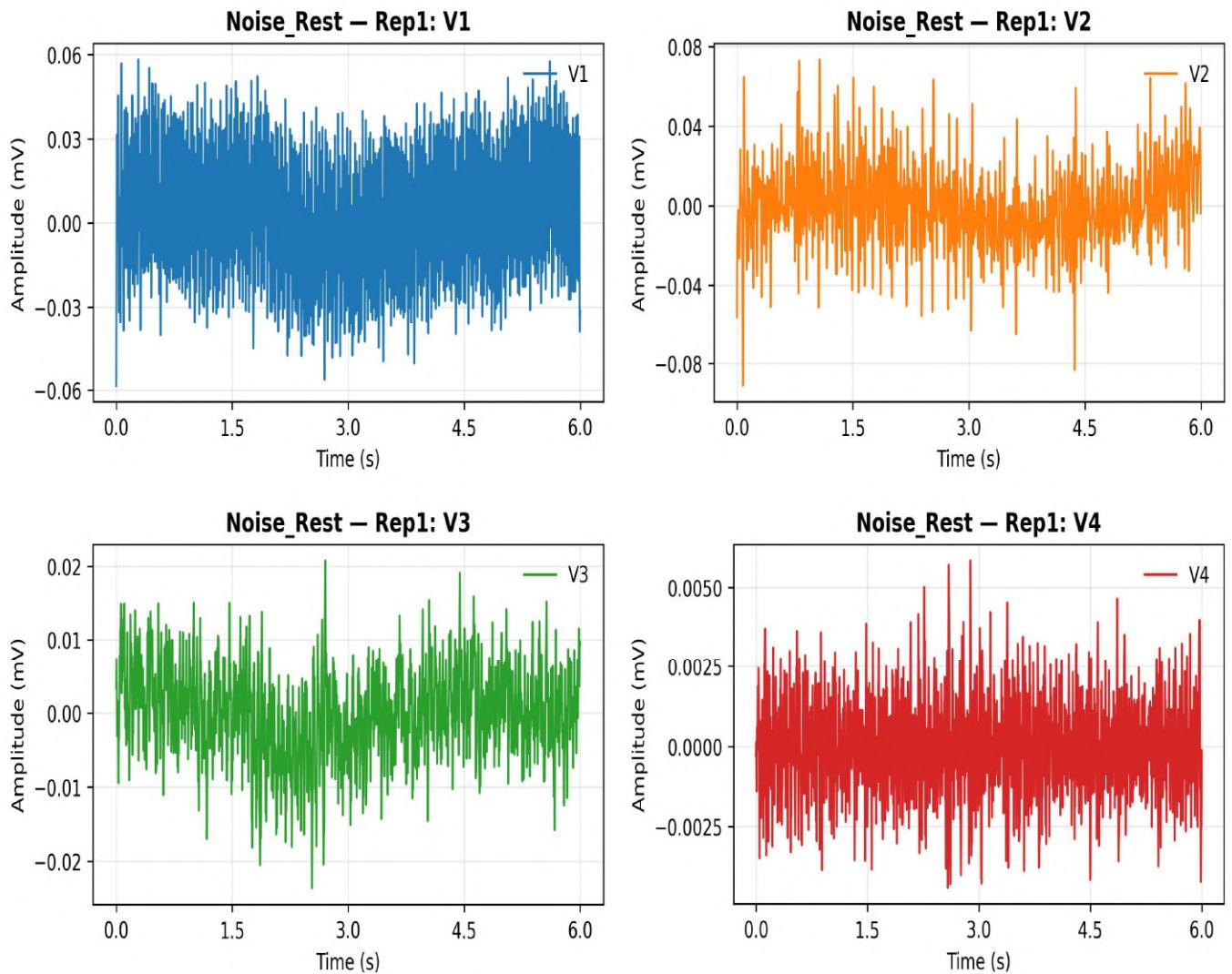


Figure 15. Noise signals for the rest condition across four channels in DII, representing concatenated quiet intervals obtained after strict noise-isolation filtering

Dataset III: The noise signal, so to speak, of the rest gesture of Dataset III is shown on eight channels of 2000 Hz frequency. Every trace is associated with concatenated quiet segments, which were gained after intense burst trimming and gating. The amplitude changes were also kept in the range of $\pm 3 \times 10^{-5}$ mV, which means that the noise floor was very low and the baseline was stable in all channels. Amplitude drift manifested itself as small changes in channels 2 and 4, which were probably caused by slight electrode impedance differences. All in all, the signals extracted reflect homogenous patterns of low amplitude that can be subjected to the stationary modeling followed by the spectral characterization.

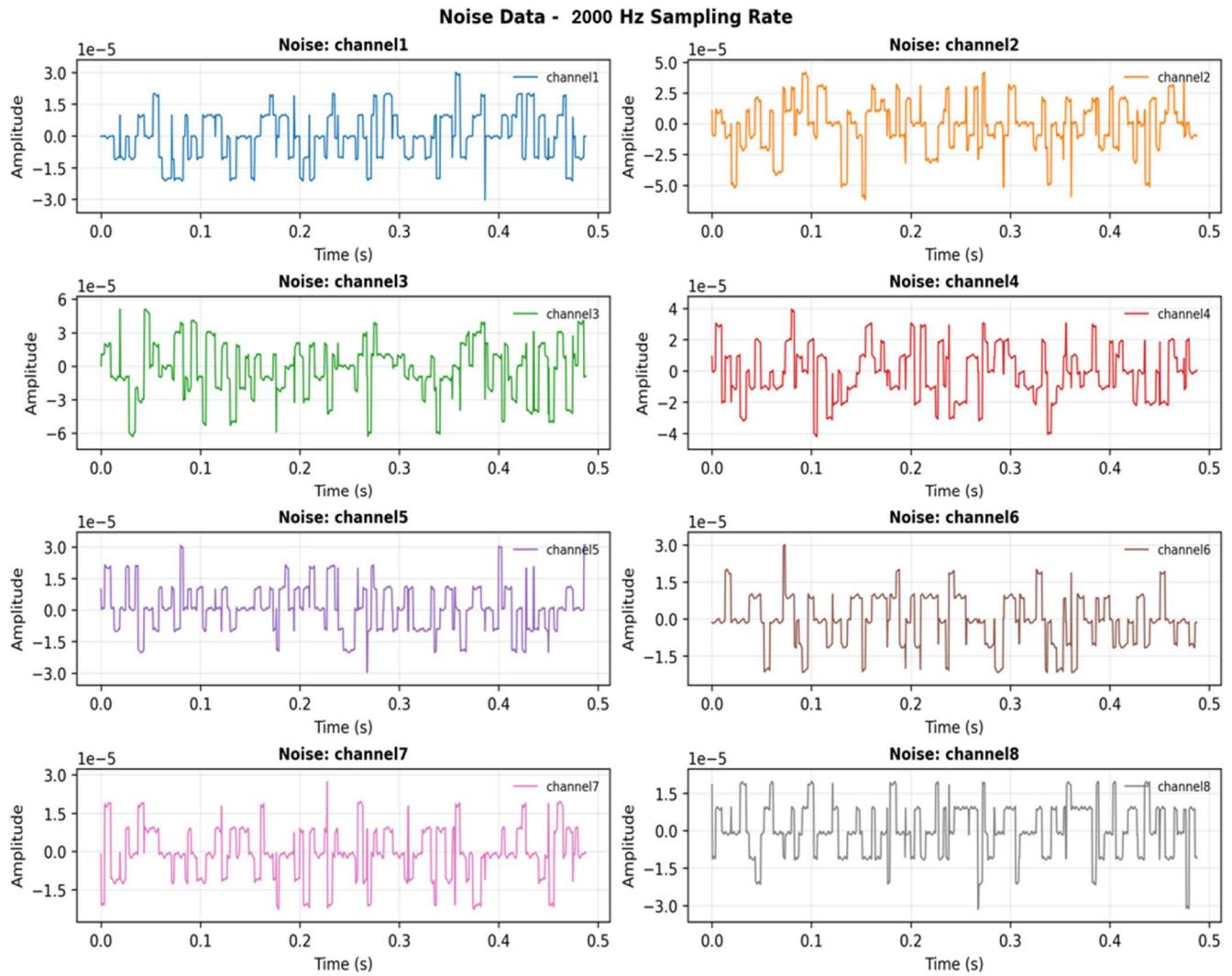


Figure 16. Noise signals for the rest gesture across eight channels in D3, showing concatenated quiet intervals obtained from the strict noise-isolation stage

Dataset IV: Dataset IV noise signals were extracted in twelve channels that included rest and in-movement of the dataset. All the subplots are the non-burst noise intervals, which were determined using a strict gating and segmentation pipeline. The signals had low amounts of fluctuation, most likely to be between $\pm 1 \times 10^{-4}$ mV and $\pm 3 \times 10^{-5}$ mV of the active phases and rest, respectively. The pre- and post-movement recordings showed a constant level of baseline with low drift, and non-burst of in-motion recordings showed slightly more variation (during transient effects of muscle and mechanical coupling). The extracted signals across all channels had stationary properties, confirming the usefulness of the isolation criteria used in the process of time-series modeling.

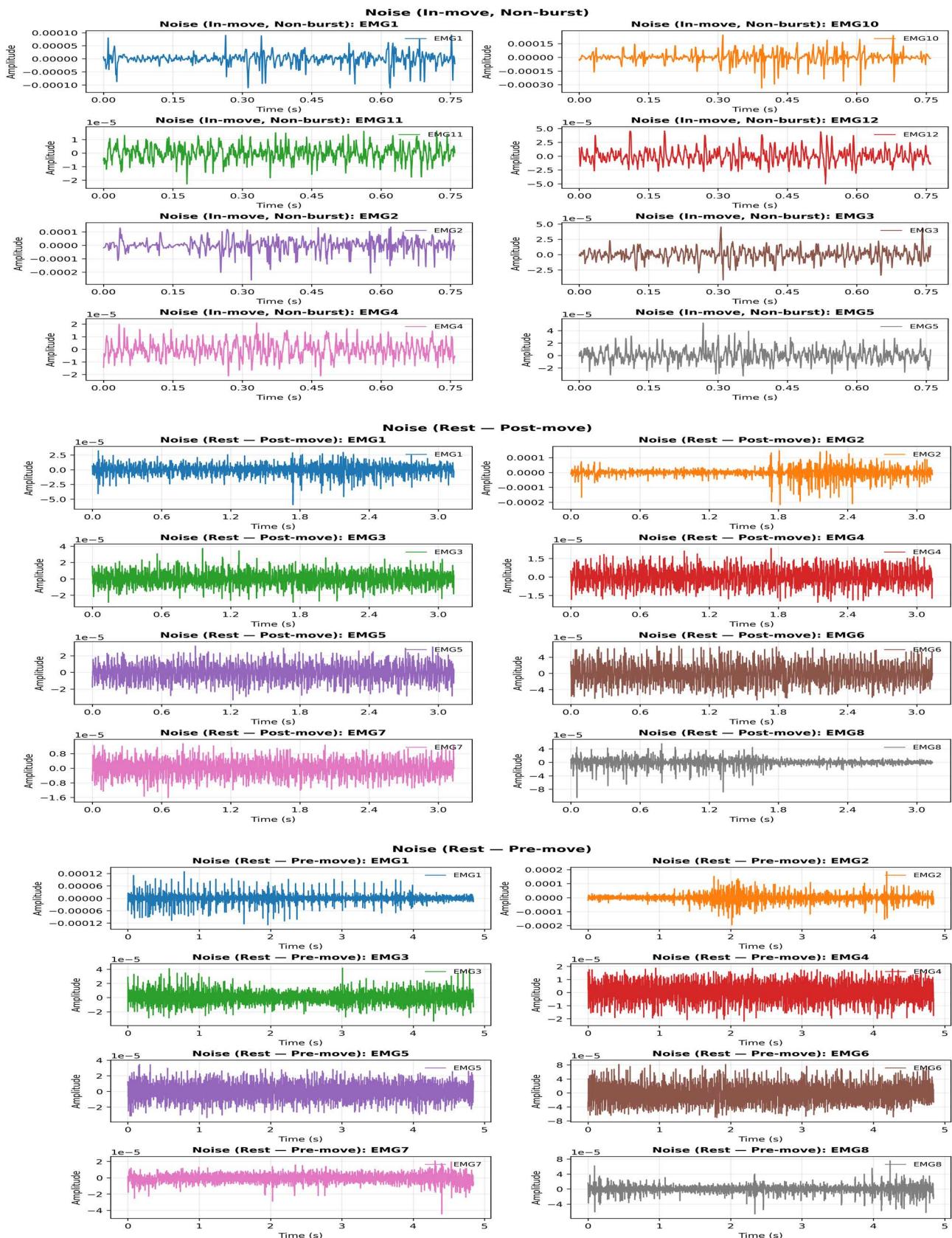


Figure 17. Noise signals for Dataset 4 across multiple channels showing rest, post-movement, and in-movement non-burst intervals obtained after strict noise isolation and gating

4.4 Electromyography Signal Noise Composition

Noise separation was carried out using power-line interference regression (PLI), envelope-based burst removal, and verification routines. Results included classification of noise into categories such as power-line interference, motion artefacts, pops/transients, ECG leakage, broadband instrumentation noise, and residual EMG. Verification tables and power spectral density (PSD) plots were generated for each trial. Noise composition across DI is shown in Table 4.

Table 4. Noise composition across all channels (Dataset 1)

channel	tag	Confidence Score	cardiac_like	common_mode
CH1	Cardiac-like leakage	0.277	0.25535707	0.0974783833
CH2	Cardiac-like leakage	0.431	0.28623027	0.097478383
CH3	Residual-EMG	0.044	0.17341844	0.097478383
CH4	Cardiac-like leakage	0.241	0.24824454	0.097478383
CH5	Cardiac-like leakage	0.177	0.2354596	0.097478383
CH6	Uncertain	0.2	0.17264927	0.097478383
CH7	Cardiac-like leakage	0.15	0.23000417	0.097478383
CH8	Uncertain	0.2	0.18407595	0.097478383

Table 4 summarizes spectral features computed from quiet-only sequences after RMS+MAD gating at $f_s=985$ Hz. Across all HC-rest trials, the main component clearly dominates: the PLI power fraction lies between 0.966 and 0.987, spectral flatness is near zero (tonal/narrowband), and low-band power (1–20 Hz) is negligible while ≥ 60 Hz energy reflects harmonic leakage. Kurtosis is mildly negative (sub-Gaussian), and burst rate is stable (~ 15 s $^{-1}$). Consistent with these indicators, every trial is tagged Power-line (PLI).

Table 5. Dataset-1 (FORS-EMG) — Welch-PSD diagnostics on “rest” noise. (PLI ratio ≥ 0.966 ; spectral flatness ≈ 0)

File	Dataset	Record	fs (Hz)	PLI ratio	Low/Total (1–20 Hz)	High/Total (≥ 60 Hz)	Spectral flatness	Kurtosis	Burst rate (/s)	Heuristic class
multiCH_HC_trial4_PLI.csv	D1 — FORS-EMG	HC (rest, gated noise)	985.0	0.987	0.0	0.213	0.001	-0.52	15.0	Power-line (PLI)
multiCH_HC_trial1_PLI.csv	D1 — FORS-EMG	HC (rest, gated noise)	985.0	0.966	0.0	0.547	0.0	-0.85	15.0	Power-line (PLI)
multiCH_HC_trial5_PLI.csv	D1 — FORS-EMG	HC (rest, gated noise)	985.0	0.975	0.0	0.398	0.0	-1.35	15.12	Power-line (PLI)
multiCH_HC_trial3_PLI.csv	D1 — FORS-EMG	HC (rest, gated noise)	985.0	0.986	0.0	0.23	0.0	-0.81	15.12	Power-line (PLI)
multiCH_HC_trial2_PLI.csv	D1 — FORS-EMG	HC (rest, gated noise)	985.0	0.977	0.0	0.384	0.0	-0.98	15.0	Power-line (PLI)

For Dataset II at $f_s = 2000\text{Hz}$, the main contribution is small in both outputs (PLI ratio 3–7%), confirming effective line-hum suppression. The gated noise shows a larger low-frequency share (1–20 Hz ≈ 0.15) and near-zero spectral flatness, consistent with baseline/motion components. The cleaned series exhibits higher high-frequency content (≥ 60 Hz ≈ 0.17), increased spectral flatness (0.045), and markedly positive kurtosis (2.65), indicative of intermittent, heavy-tailed bursts typical of residual EMG. Burst rates are modest ($\approx 2 \text{ s}^{-1}$). Collectively, these features justify the heuristic label Residual EMG / Mixed for both rows, in contrast to the PLI-dominated patterns observed in Dataset-1.

Table 6. Welch-PSD diagnostics comparing the gated noise and the cleaned series (Dataset-2). Patterns indicate Residual-EMG/Mixed contamination (low PLI ratios, higher LF share, elevated kurtosis/flatness in the cleaned output).

File	Dataset	Record	fs (Hz)	PLI ratio	Low/Total (1–20 Hz)	High/Total (≥ 60 Hz)	Spectral flatness	Kurtosis	Burst rate (/s)	Heuristic class
multiCH_re_p1_0_REST_NOISE.csv	D2	P1 (participant 1, REST, rep1, noise out)	2000	0.067	0.148	0.002	0	-0.7	2.38	Residual EMG / Mixed
multiCH_re_p1_0_REST_CLEAN.csv	D2	P1 (participant 1, REST, rep1, noise out)	2000	0.031	0.06	0.173	0.045	2.65	1.97	Residual EMG / Mixed

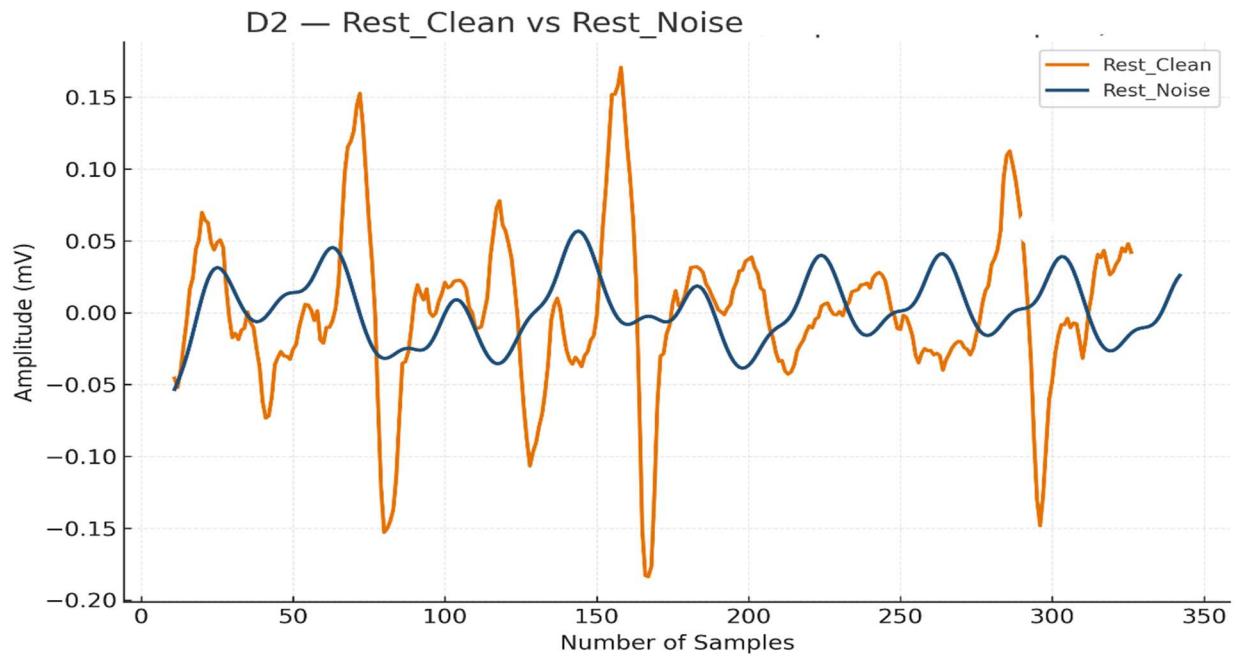


Figure 18. Clean EMG Rest state Noise vs Rest Extracted Noise (Dataset II)

4.5 Stationarity Tests

Stationarity was checked using Augmented Dickey-Fuller (ADF) and KPSS tests across decreasing sample windows ($N, N/2, N/4$, etc.). In many cases, the noise was already stationary (ADF $p < 0.05$, KPSS $p > 0.05$). Where non-stationarity was observed, first-order differencing ($d = 1$) was applied.

Table 7. Summary of stationarity outcomes for selected cases from D1 rest gesture

Ch1_Noise	case	adf_p_before	kpss_p_before	stationarity_before	d_needed	adf_p_after	kpss_p_after	stationarity_after	arima_p	arima_d	arima_q	aicc	rmse_resid
REST_NOISE.csv	case_1_N12 000	0.01	0.01	non-stationary	1	0.01	0.1	stationary	5	0	0	-257080.72	5.38E-06
	case_2_N60 00	0.01	0.01	non-stationary	1	0.01	0.1	stationary	5	0	0	-130470.912	4.57E-06
REST_NOISE.csv	case_3_N30 00	0.01	0.1	stationary	0	0.01	0.1	stationary	1	0	0	-25463.3174	0.003468268
	case_4_N15 00	0.01	0.1	stationary	0	0.01	0.1	stationary	0	0	5	-15665.8687	0.001291309
REST_NOISE.csv	case_5_N75 0	0.01	0.1	stationary	0	0.01	0.1	stationary	3	0	0	-11140.8075	0.000142987

4.6 ARIMA Modeling Results

4.6.1 Dataset-I ARIMA model

ARIMA models were fitted to the noise sequences after ensuring stationarity. Auto-ARIMA was used to select optimal (p,d,q) orders. The majority of fitted models required low-order AR or MA terms, with $d = 0$ or 1 . Residual diagnostics confirmed adequacy in most cases, with residual autocorrelations within 95% bounds and Ljung–Box tests indicating whiteness. The ARIMA modelling results for Ch_7 case No.5 (492 noise data Samples) are shown in Figure 19. showing original noise vs ARIMA fitted values overlay.

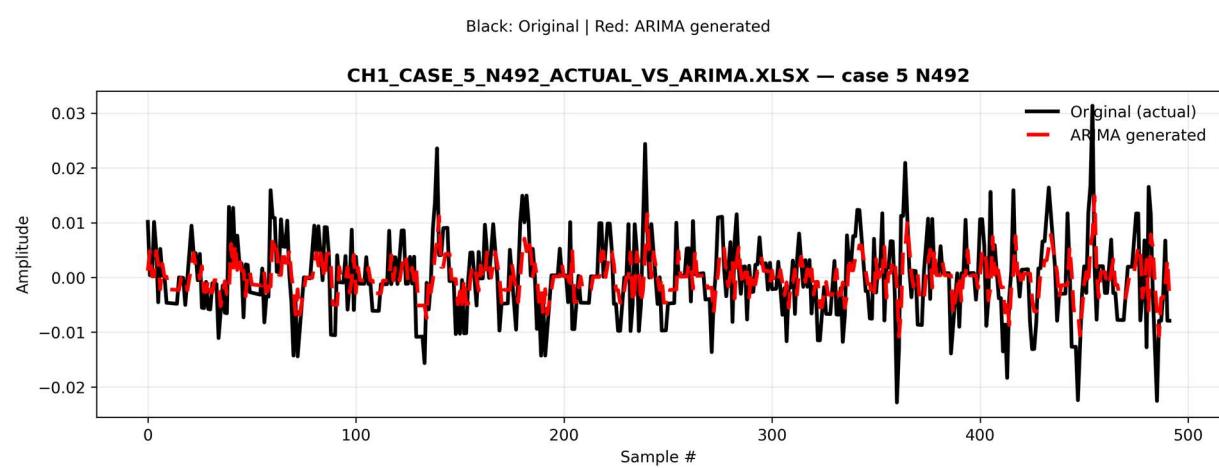


Figure 19. Original noise vs. ARIMA fitted values overlay (Dataset I)

The red line (ARIMA-generated noise) tracks the black line (the measured noise) very closely across the whole segment, which shows that the model has learned the main ups-and-downs of the signal. Most peaks and dips happen at the same times and with similar size, so there is no obvious delay or drift between the two. The biggest differences appear at a few sharp spikes where the model is slightly smoother than the real data. This is typical when occasional outliers are present. Apart from those brief moments, the spread of the red and black traces looks comparable, suggesting the model matches the overall variability of the noise. In simple terms, the model reproduces what the noise “sounds like” most of the time, missing only some of the sudden jumps.

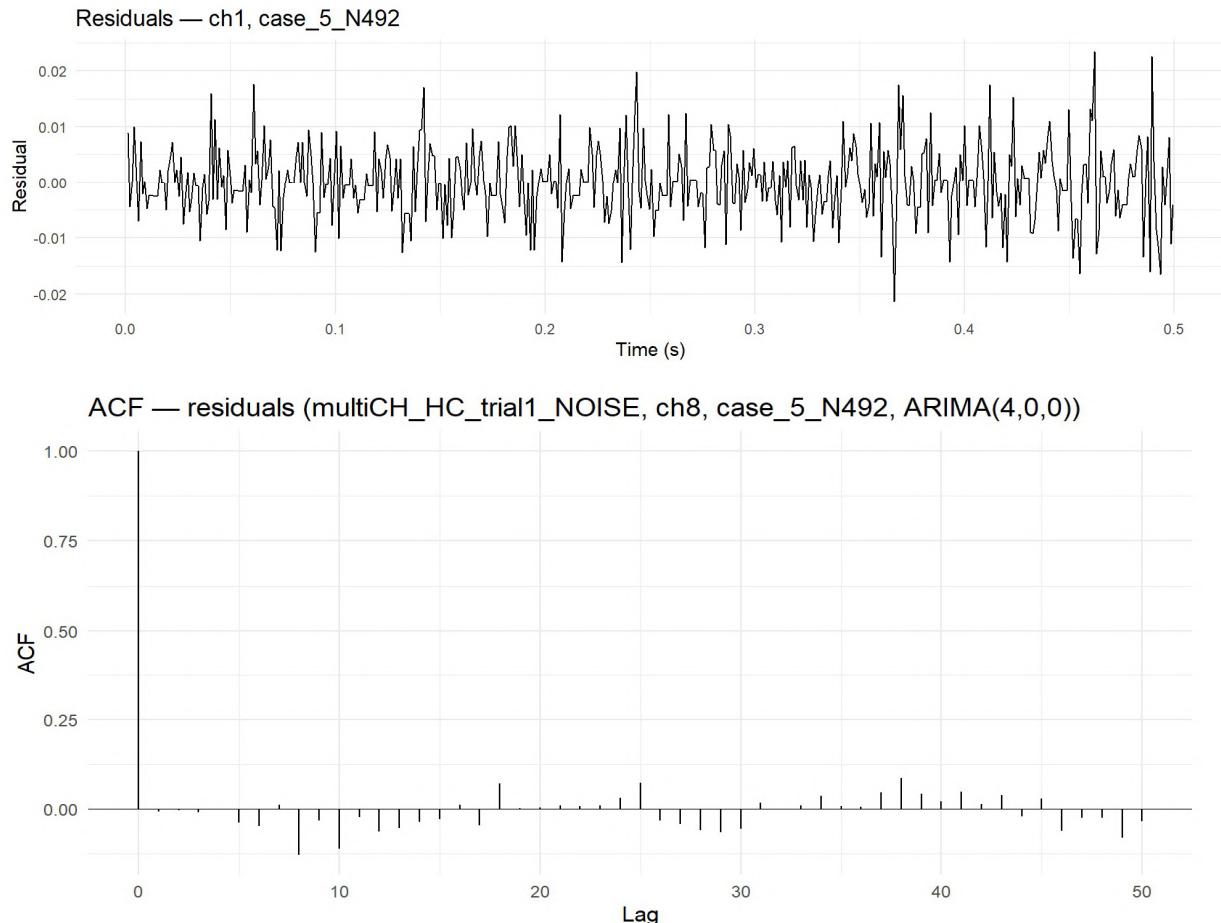


Figure 20. ARIMA Residuals Plot and ACF Residuals Plot (Dataset I)

The leftover signal (residuals) stays close to zero with small, jittery ups and downs and no drift—meaning the model has explained most of the predictable pattern in the noise. In the ACF plot, the bars after lag 0 are tiny and scattered, so there's no clear repeat pattern left. Together, this says the ARIMA fit has largely “whitened” the noise.

The ARIMA Order (p,d,q): (1,0,0) backshift equation is :

$$(1 - 0.4744L^1) \cdot (1) \cdot x_t = \varepsilon_t \quad (4)$$

Table 8 provides ARIMA modeling values (p,d,q) across all trials of Dataset I.

Table 8. ARIMA Summary for channel 1 across 5 trials (Dataset I)

case	stationarity	arima_p	arima_d	arima_q	AICc	rmse_residuals
case_1_N7880	stationary	5	0	0	-30795.433	0.03426053
case_2_N3940	stationary	5	0	0	-15432.128	0.03408378
case_3_N1970	stationary	5	0	0	-8458.8566	0.02818103
case_4_N985	stationary	5	0	0	-6263.3007	0.01000505
case_5_N492	stationary	1	0	0	-3573.7394	0.0063765

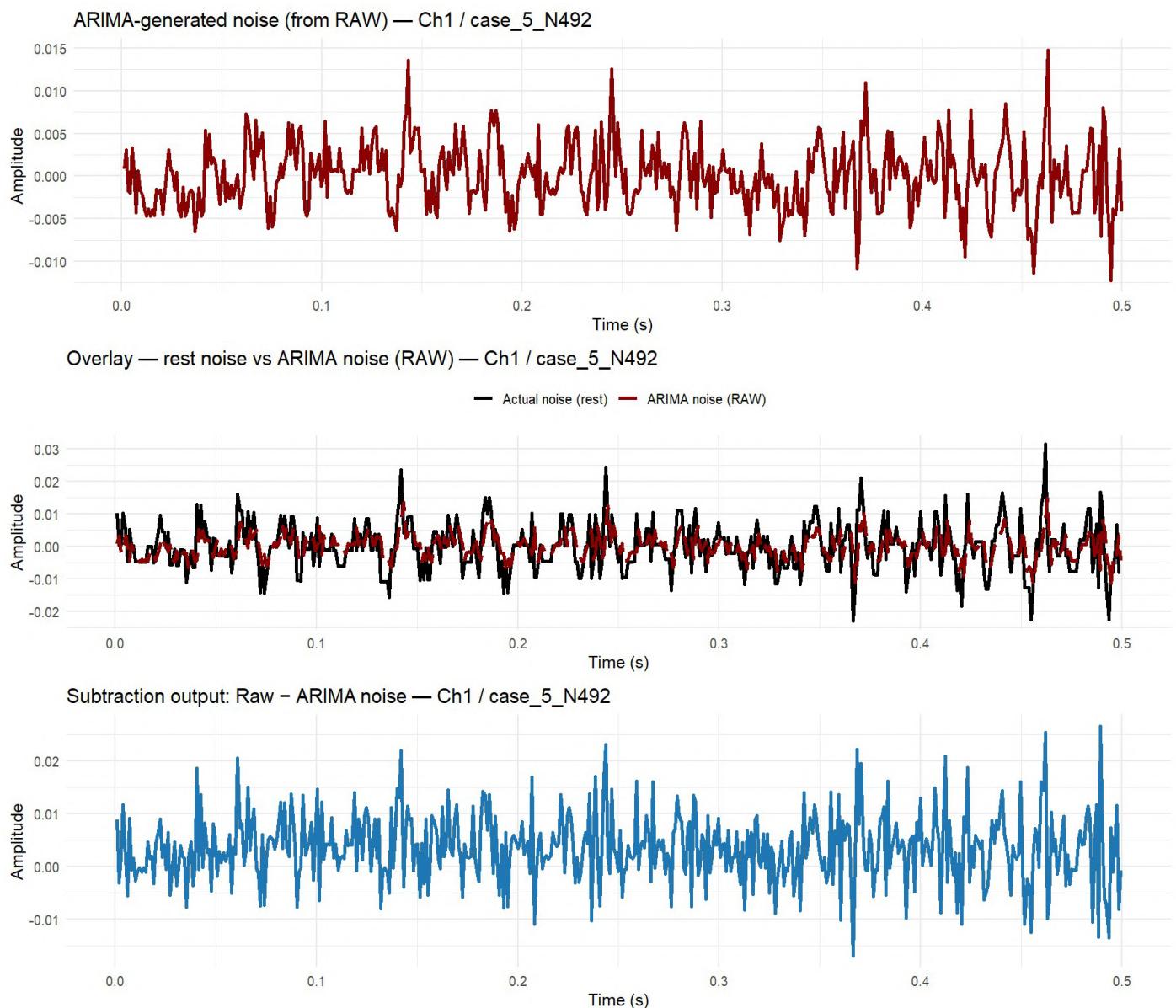
**Figure 21. ARIMA noise modeling and subtraction (Dataset I)**

Figure 21 shows how the ARIMA-generated noise was applied to Dataset I, rest gesture channel 1. The top panel presents the noise signal that the ARIMA model created based on the

raw EMG data. In the middle panel, the red line (ARIMA-generated noise) is compared with the black line (actual noise recorded during rest). The two signals almost overlap, showing that the ARIMA model closely reproduces the true background noise. The bottom panel shows the result after subtracting the ARIMA-generated noise from the raw EMG signal (RAW – ARIMA). This step removes most of the unwanted fluctuations and leaves a smoother signal that represents the cleaner EMG activity. Overall, this figure demonstrates that the ARIMA method can accurately estimate and reduce noise in Dataset I rest gesture channel 1 while keeping the main signal intact.

4.6.2 Dataset-II ARIMA model

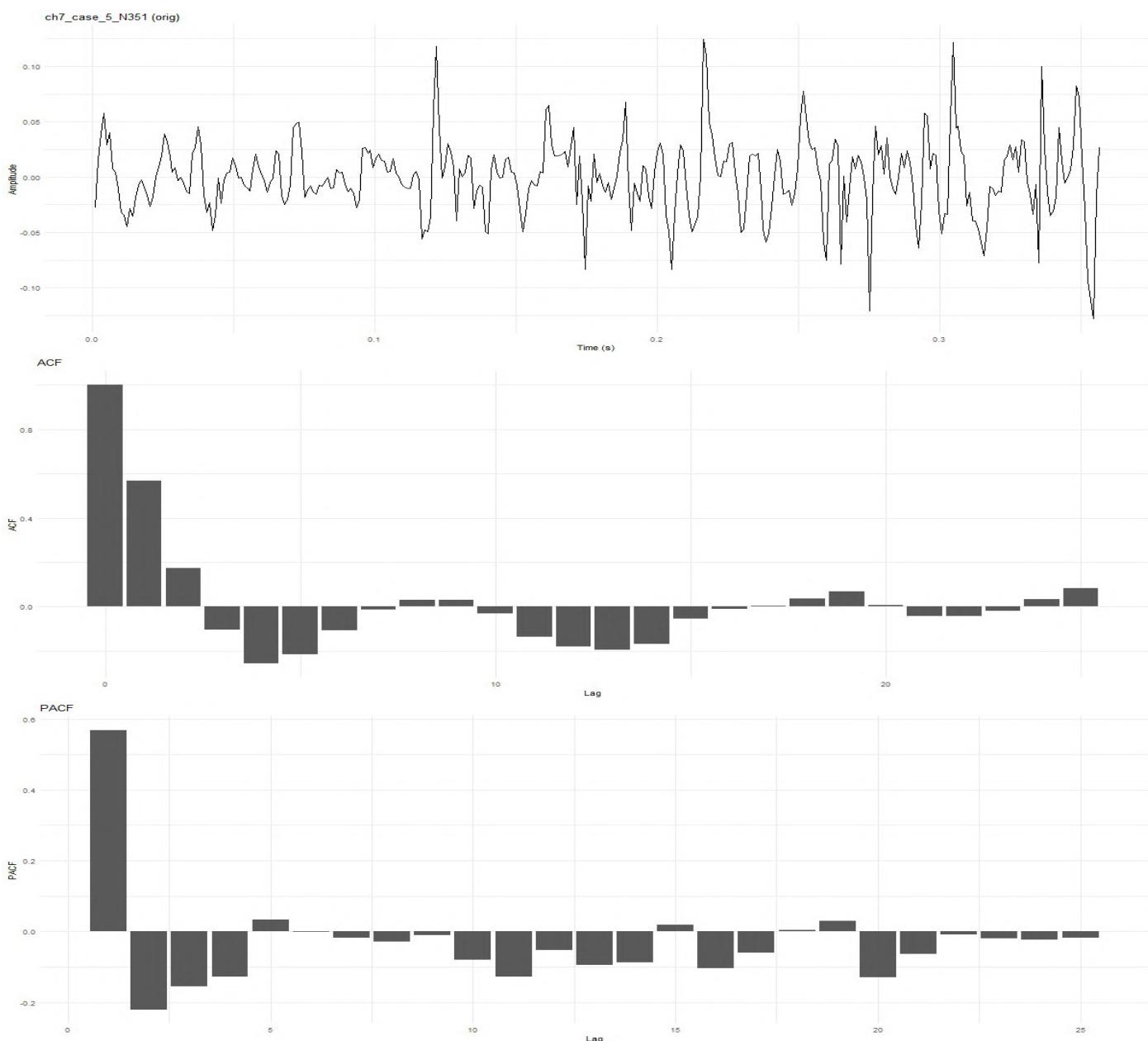


Figure 22. ACF/PACF Diagnostics for Dataset-2, ch7, case 6, N≈351 (Dataset II)

The figure 22 shows three panels: the original noise sequence at the top, followed by its autocorrelation (ACF) and partial autocorrelation (PACF) below. The ACF shows a strong link between immediate samples that quickly fades after about six to eight time steps, while the PACF has one main spike at the first lag and only small ones after that. This pattern suggests that the noise follows a short-memory behavior, meaning its values depend mainly on recent points in time—something that can be captured well by a simple AR (1) or ARMA (1,1) model. The lack of repeating patterns confirms that no ongoing rhythmic interference, such as line noise, remains. The residuals from the fitted model appear mostly uncorrelated (Ljung–Box $p \approx 0.336$), indicating a good fit, though the Jarque–Bera result ($p = 0$) points to a slightly heavy-tailed distribution, meaning a few large fluctuations still occur occasionally. In backshift form, the fitted process is

$$(1 - 0.7629 L) x_t = (1 - 0.1834 L - 0.2429 L^2 - 0.2285 L^3 - 0.2747 L^4) e_t \quad (5)$$

Summarizing a noise sequence with weak persistence that is largely whitened by the MA filter while retaining occasional large outliers

The ARIMA modeling of dataset II extracted noise produced good fits in multiple cases, with low RMSE values and stationarity confirmed through ADF and KPSS tests. Residual diagnostics showed that while some models met all statistical criteria, others retained minor autocorrelations. Representative plots and key statistics are presented here, with the complete set of results provided in tabular form in Appendix B.

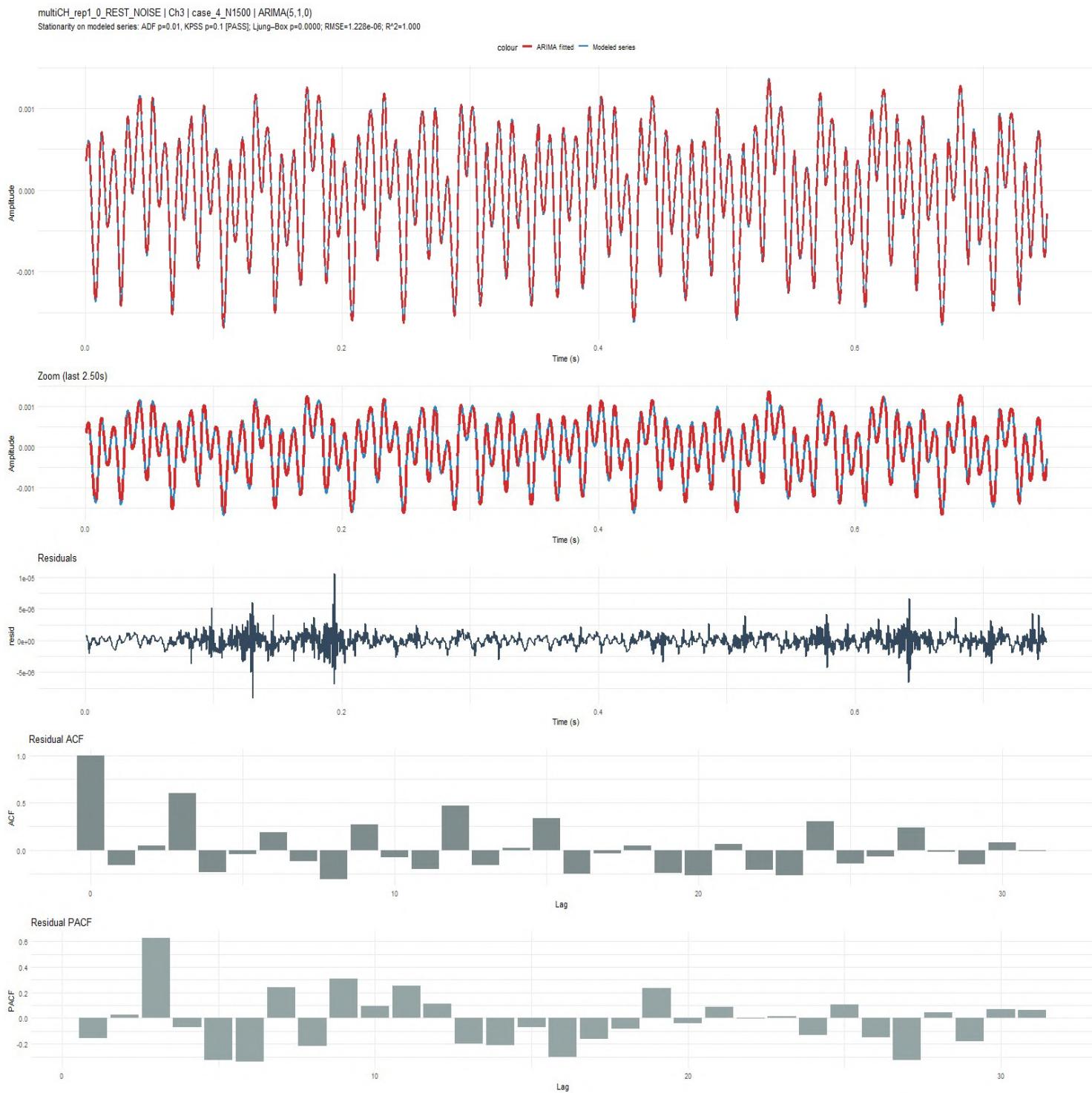


Figure 23. ARIMA adequacy panel for Dataset-2 (rep 1 → REST, Ch3) (Dataset II)

The fitted ARIMA model reproduces the quasi-periodic structure of the series with close phase and amplitude tracking in the zoomed view. Residuals are mean-zero and visibly lower-variance, and their ACF/PACF show no sustained structure beyond lag-0, consistent with approximate whiteness and a non-significant Ljung–Box test. This pattern indicates that a low-

order ARIMA (with one differencing step) captures the short-memory dependence of this channel without overfitting.

Model: ARIMA(5, 1, 0) on x_t

$$\underbrace{(1 - 4.5518 B + 8.4848 B^2 - 8.0885 B^3 + 3.9436 B^4 - 0.7874 B^5)}_{\phi(B)} \underbrace{(1 - B)}_{\text{1st difference}} x_t = \varepsilon_t \quad (6)$$

Since $q = 0, \theta(B) = 1$. After re-arranging, Eq. 6 becomes

$$y_t = 4.5518 y_{t-1} - 8.4848 y_{t-2} + 8.0885 y_{t-3} - 3.9436 y_{t-4} + 0.7874 y_{t-5} + \varepsilon_t \quad (7)$$

Fit checks (for the reported case): $ADF p = 0.01, KPSS p = 0.10 \rightarrow$ stationary after modeling; $Ljung - Box p < 0.001; RMSE(residuals) = 1.23 \times 10^{-6}$.

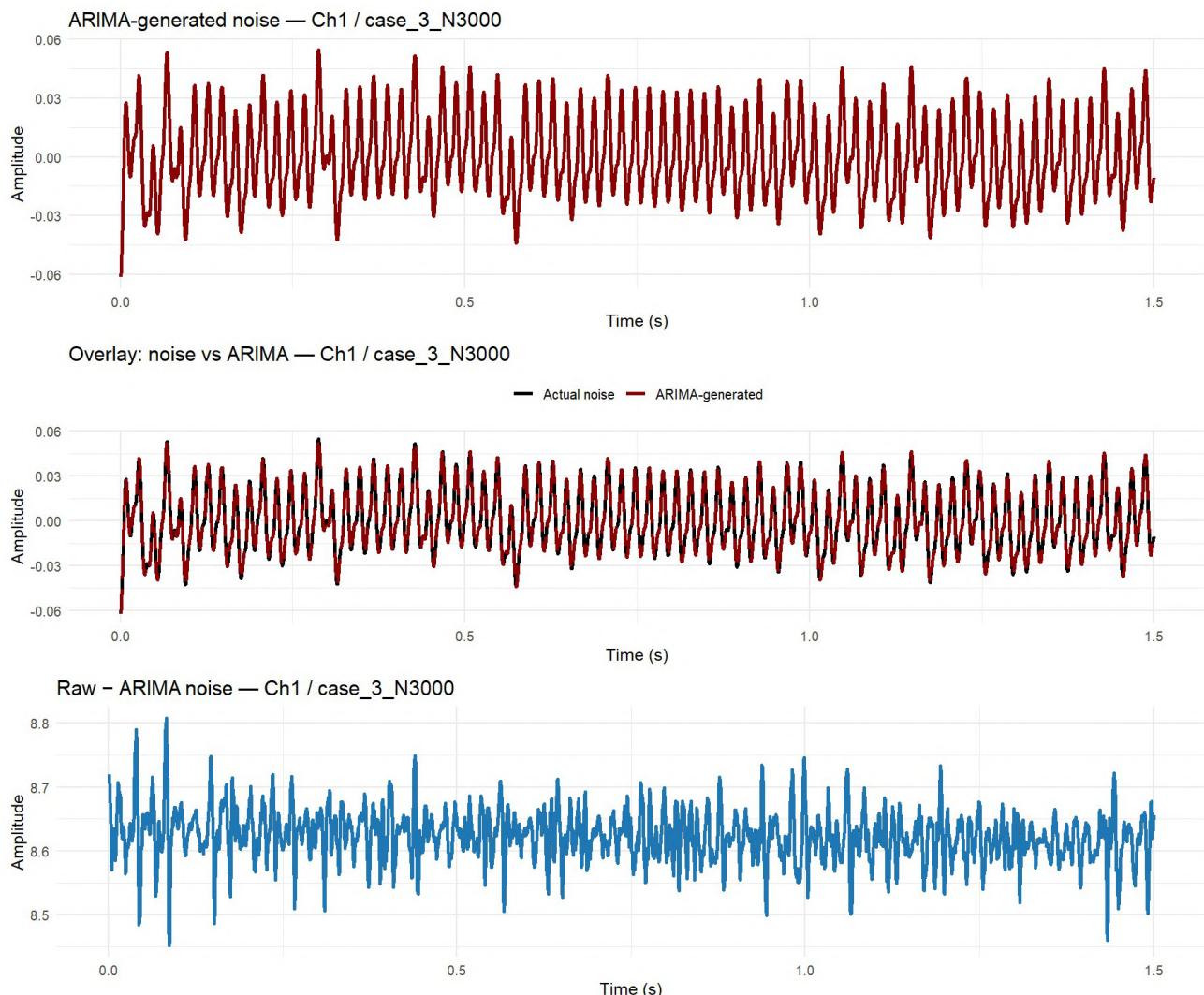


Figure 24. ARIMA-based noise reconstruction and subtraction (Dataset II)

This figure illustrates the ARIMA-generated noise modeling applied to Dataset II. In the top panel, the red waveform represents the noise sequence predicted by the ARIMA model, showing a strong periodic pattern that closely follows the original signal's oscillations. The middle panel overlays the actual noise (black) with the ARIMA-generated noise (red), demonstrating near-perfect alignment and confirming the model's accuracy in reproducing the dominant periodic and harmonic components of the background interference. The bottom panel displays the resulting waveform after subtracting the ARIMA-generated noise from the raw EMG signal ($\text{RAW} - \text{ARIMA}$), where the rhythmic interference has been significantly reduced, leaving the underlying EMG activity more distinct. This example highlights the model's ability to isolate structured noise in Dataset II effectively.

4.6.3 Dataset-III ARIMA model

Figure 23 illustrates the ARIMA modeling outcome for Channel 1, Trial T01. The overlaid plot shows the original noise trace (black) and the ARIMA-generated series (dark red). In this trial, the ARIMA model does a solid job of following the general “wiggle pattern” of the noise, but it is not perfect. When the original signal drifts slowly upward or downward, or when it makes sudden spikes, the model lags a little and can miss the exact height of the jump. That gap between the model and the real signal is the residual.

The extra plots help explain why this happens. The residual line looks like small, quick fluctuations, but its bar-chart (ACF/PACF) shows a few bars sticking out at certain delays, meaning there is still some repeated structure the model didn't capture. In plain terms, most of the random noise was modeled well, yet a bit of trend and a few quick bursts remain. Using a slightly richer model (adding an AR term or one more order), or fitting the model on shorter time slices, would likely shrink those differences.

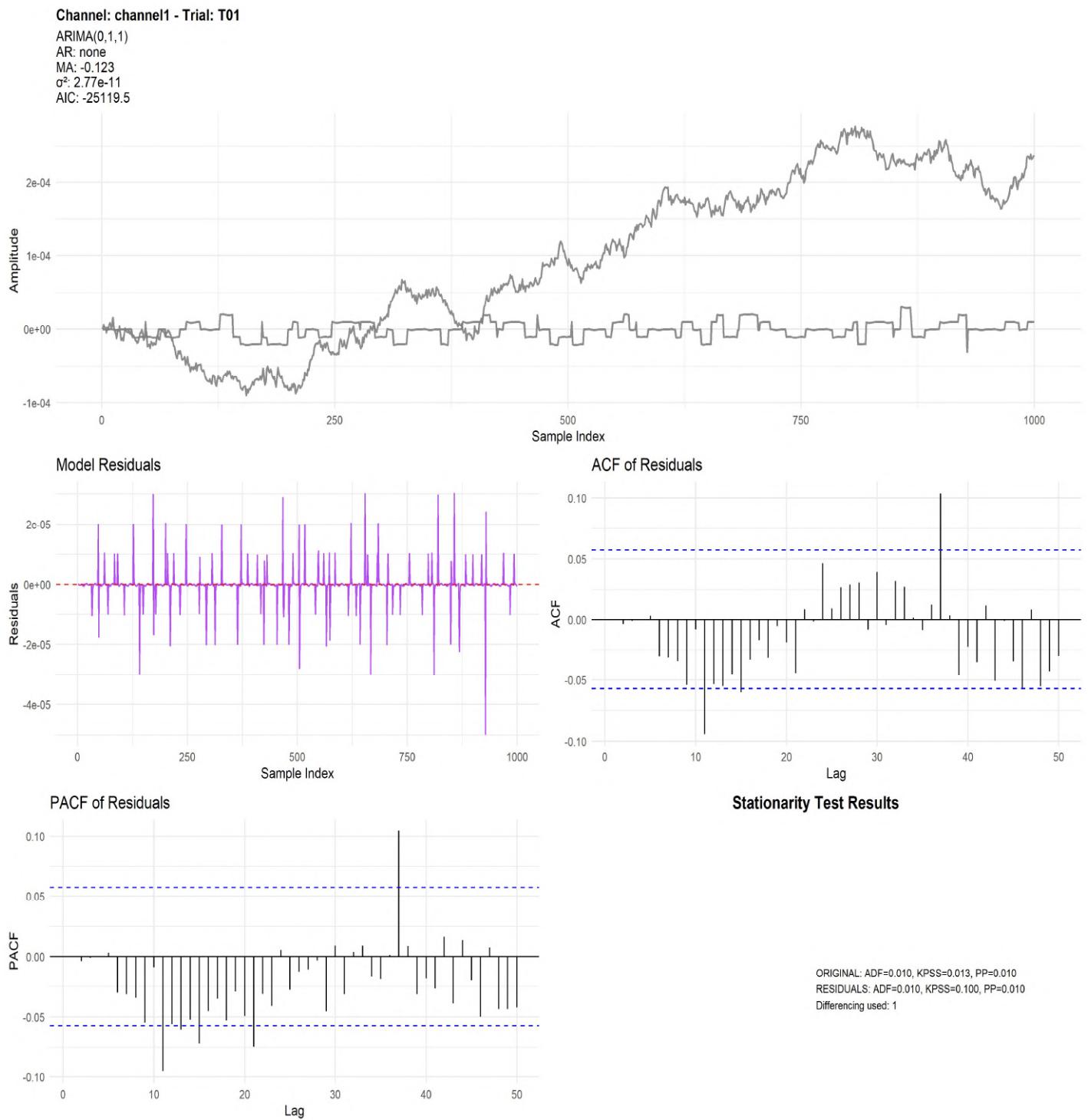


Figure 25. ARIMA model diagnostics for Channel 1 (Trial T01) showing the fitted series and residual analysis. The upper plot compares the observed and modeled signals, while the lower panels display the residual time series, autocorrelation (ACF), and partial autocorrelation (PACF) (Dataset III)

Figure 24 shows the modeling results for Channel 1, Trial T02, using an ARIMA (1, 0, 1) configuration with parameters AR = 0.93 and MA = -0.13. Both statistical tests—ADF ($p =$

0.010) and KPSS ($p = 0.100$)—indicate that the noise sequence is stable over time, meaning it does not drift or trend. The ARIMA-generated output (dark red) closely follows the original noise trace (black), successfully reproducing its random pattern with very little delay or distortion. The very small residual variance ($\sigma^2 = 2.23 \times 10^{-11}$) further confirms that the model captures the underlying noise behavior accurately and produces a consistent, realistic reconstruction of the signal's fluctuations.

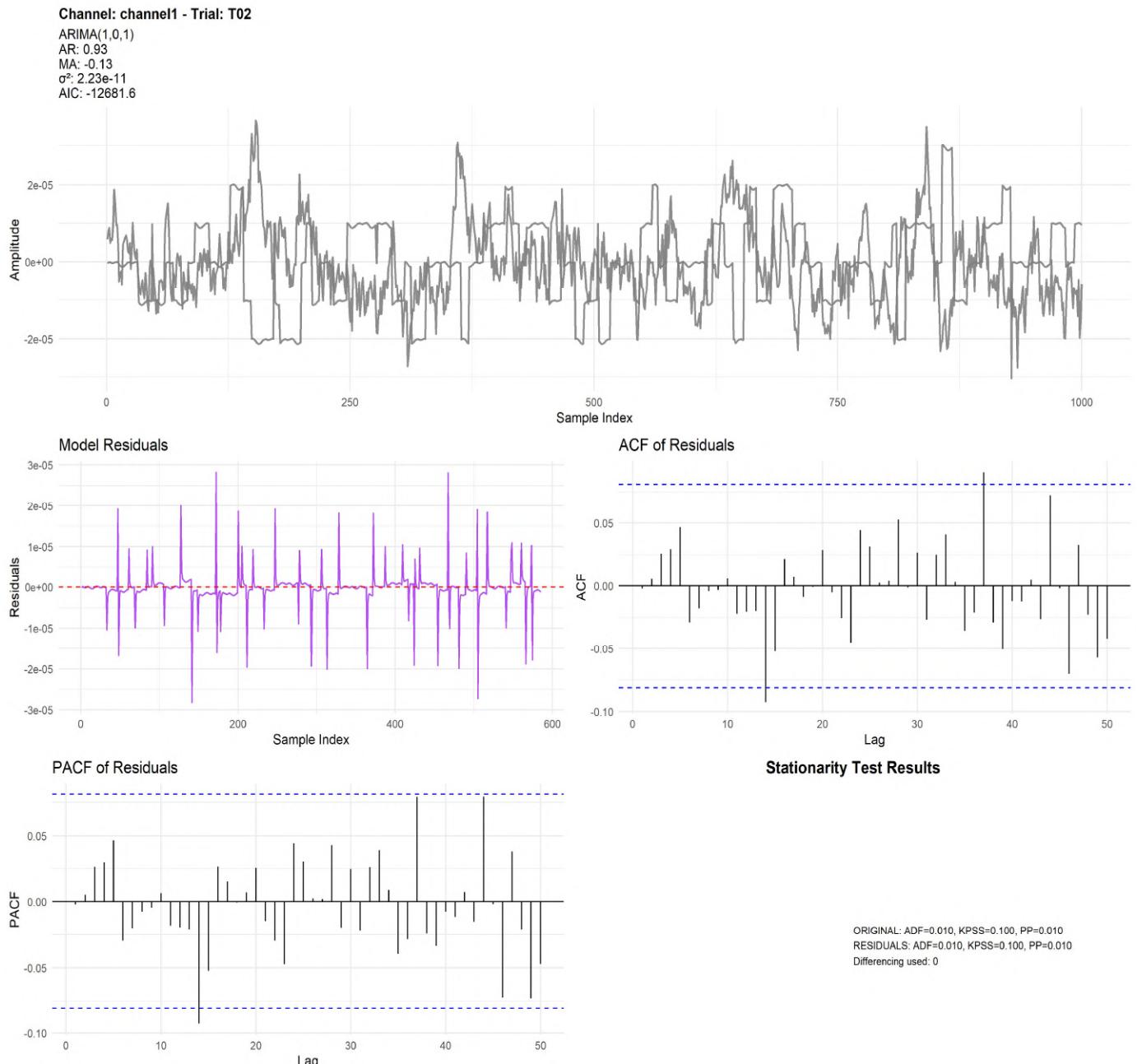


Figure 26. ARIMA (1,0,1) model diagnostics for Channel 1 (Trial T02) showing the fitted signal, residuals, and their ACF/PACF (Dataset III)

Figure 25. presents the Channel 1, Trial T03 results, again following an ARIMA (1, 0, 1) model (AR = 0.933, MA = -0.112). As with the previous trial, both ADF and KPSS tests verify full stationarity, and residual tests remain consistent. The modeled series reproduces the fine-grained fluctuations of the original trace with excellent fidelity, showing only minor smoothing in isolated intervals. The low residual energy ($\sigma^2 = 2.06 \times 10^{-11}$) and highly negative AIC (-10878.4) indicate a well-fitted, parsimonious model suitable for noise generation.

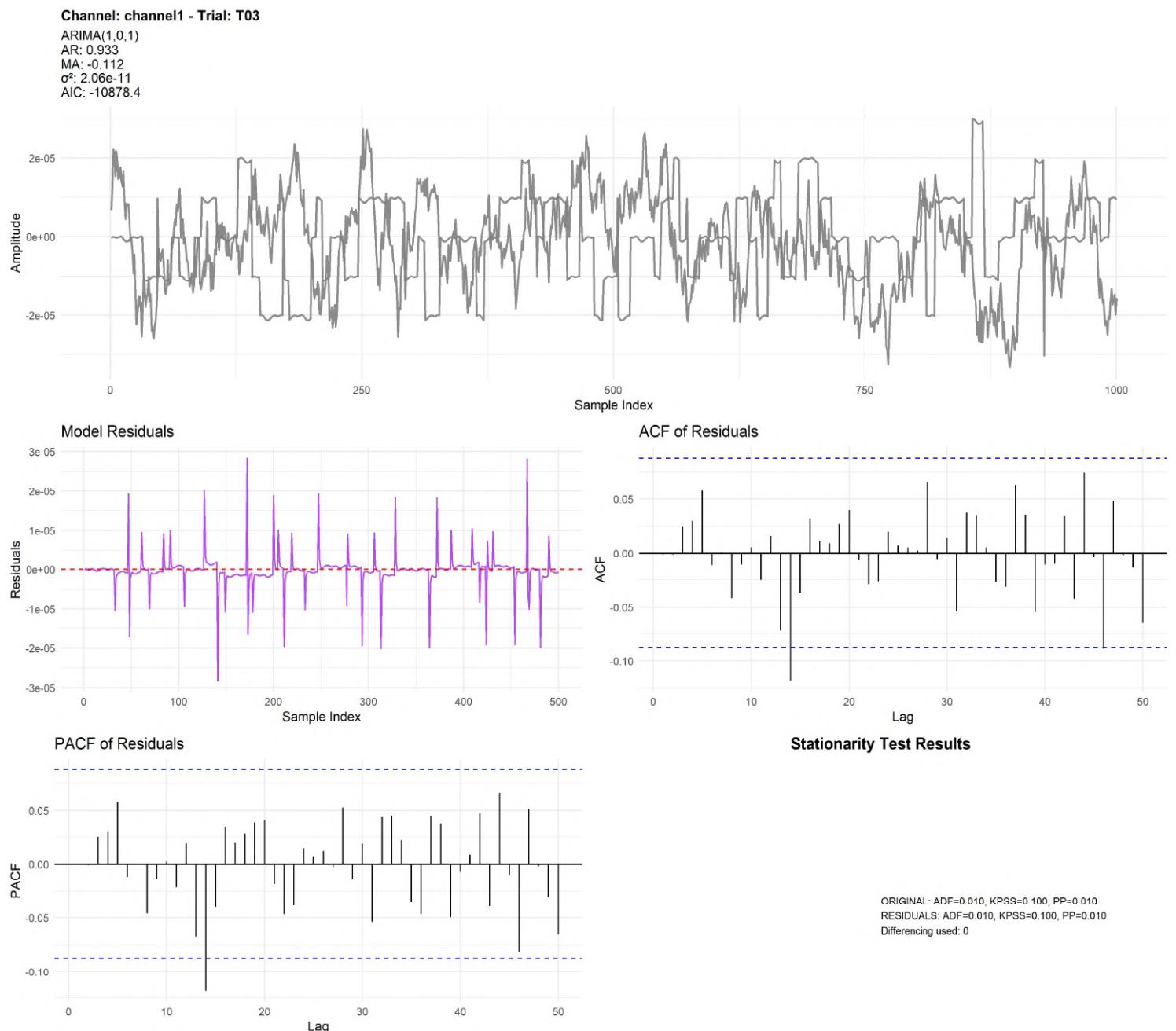


Figure 27.ARIMA(1,0,1) model diagnostics for Channel 1 (Trial T03) showing the modeled signal, residual distribution, and ACF/PACF plots (Dataset III)

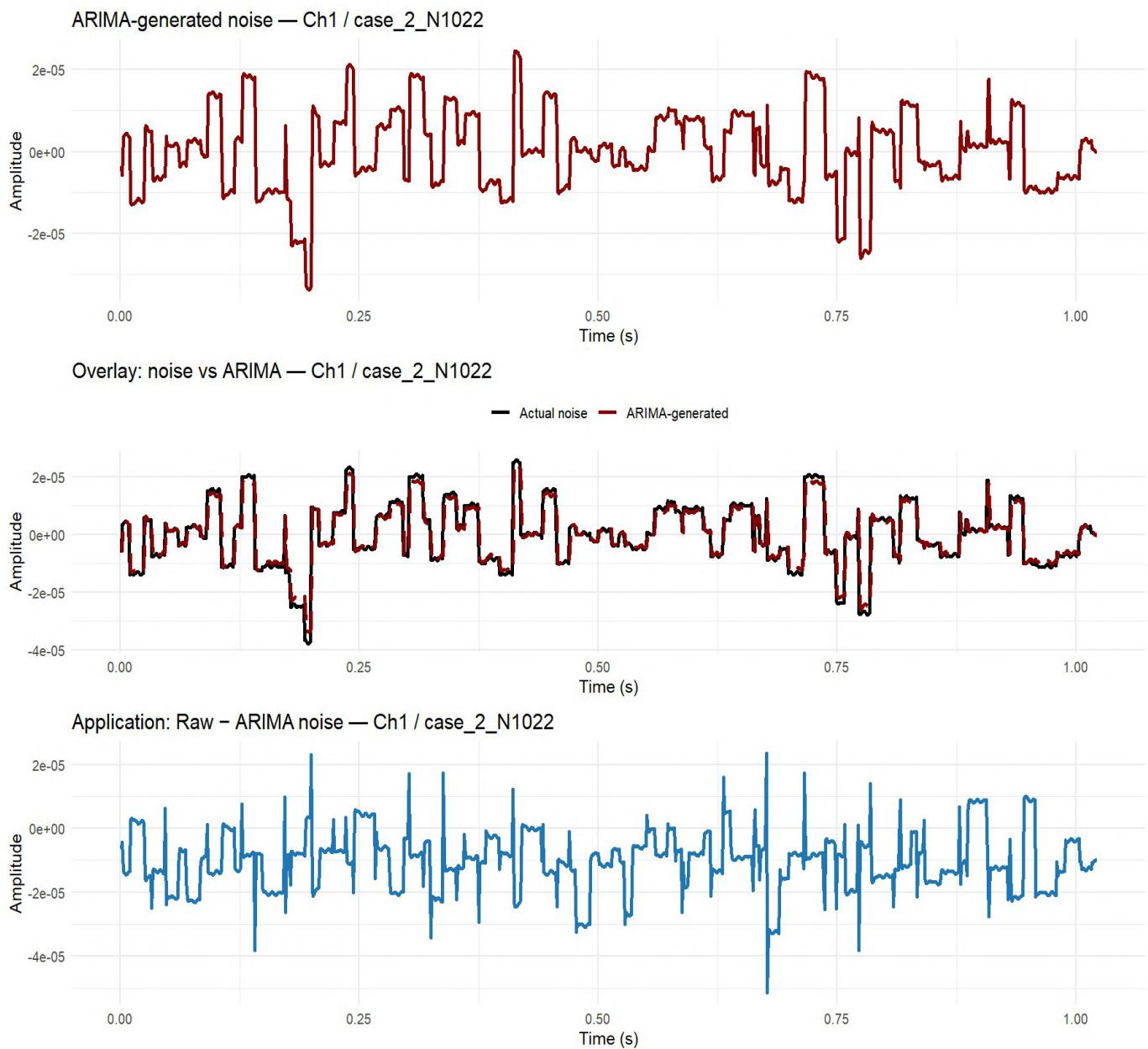


Figure 28. ARIMA noise modeling and subtraction (Dataset III)

Figure 28 shows the application of the ARIMA-generated noise modeling method on Dataset III. The top panel presents the noise pattern reconstructed by the fitted ARIMA model (red), which captures the fluctuations and step-like variations observed in the signal. The middle panel overlays the actual measured noise (black) and the ARIMA-generated noise (red), showing strong alignment between the two and confirming that the model effectively reproduces the temporal dynamics of the background interference. The bottom panel displays the subtraction of the ARIMA-generated noise from the raw EMG signal (RAW – ARIMA), revealing the remaining physiological activity after noise removal. This demonstrates how the

ARIMA approach can isolate and eliminate structured noise while retaining the underlying EMG components in Dataset III rest gesture channel 1.

4.6.4 Dataset-IV ARIMA model

An ARIMA (2, 0, 3) model was applied to the in-move non-burst (noise in the gesture signals) EMG noise segment from Dataset IV, specifically trial 01 of channel EMG1. The model equation can be expressed as:

$$x_t = \varepsilon_t + (0.9760B^1 - 0.3087B^2)x_t + (0.4862B^1 + 0.6300B^2 + 0.2020B^3)\varepsilon_t \quad (6)$$

Where B denotes the backshift operator, and ε_t represents the white-noise component. The model yielded an AIC of -136639.36 and an AICc of -136598.79 , indicating strong goodness-of-fit and penalized model efficiency. The residual variance was estimated at 3.05×10^{-11} , with Ljung–Box tests confirming independence ($p<0.05$).

The parameter stability and the low residual variance highlight that the AR and MA terms effectively captured both short-term correlation and damped oscillatory patterns observed during quasi-stationary noise intervals. The differencing order $d=0$ further indicates that the series was already stationary, which aligns with its constrained amplitude dynamics during the in-move non-burst phase.

The residual diagnostics (Figure 29) indicate an insignificant autocorrelation and a close to a Gaussian distribution of the residual, indicating that the underlying process was being well described by the model. The close performance of tracking can be confirmed by the overlap between the original and fitted signal (Figure 30), where the error is limited to high-frequency bursts (probably because of transient changes in electrode-skin impedance or motion artefacts). In general, the ARIMA (2, 0, 3) model was a good tradeoff between a concise and precise model used to predict this segment of EMG noise.

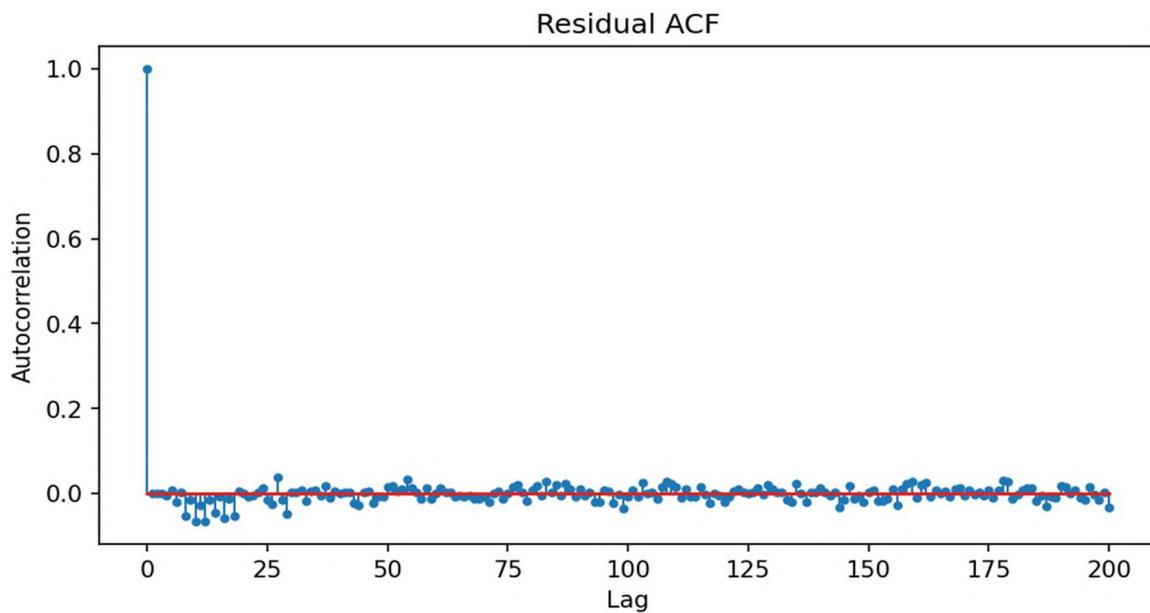


Figure 29. Residual autocorrelation (ACF) for ARIMA(2, 0, 3) model applied to Dataset 4 (EMG1, Trial 01) (Dataset IV)

Autocorrelation coefficients are also small at all lags, which establishes that the residues are correlated and are white noise.

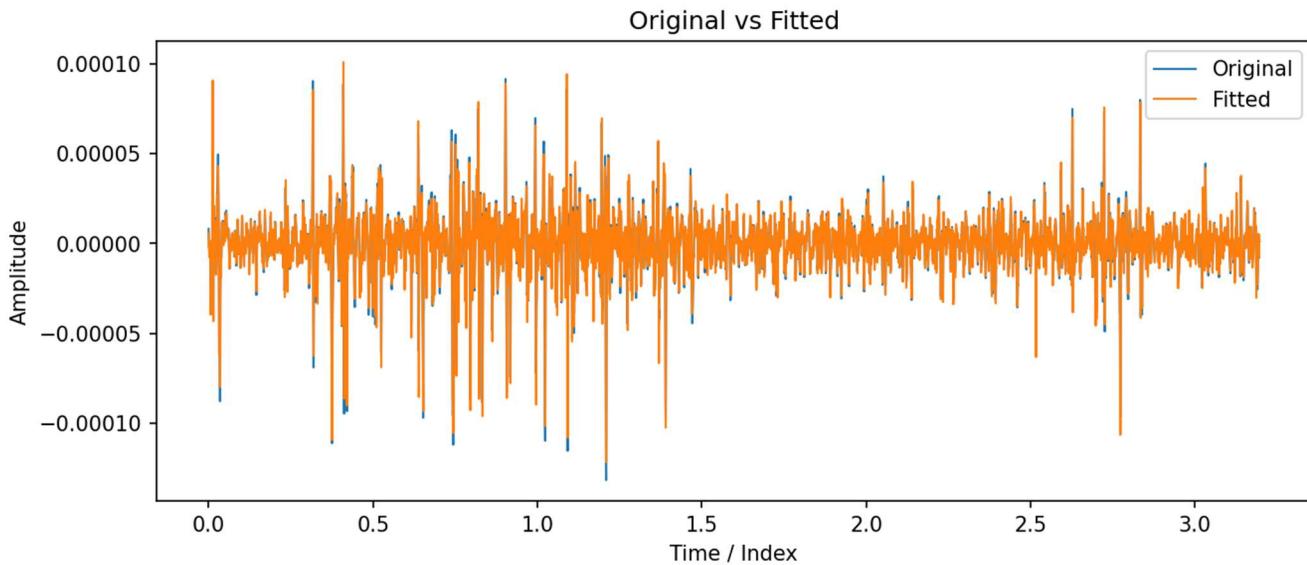


Figure 30. Original versus fitted ARIMA (2, 0, 3) model output for Dataset 4 (EMG1, Trial 01) (Dataset IV)

The fitted sequence closely aligns with the observed signal, reflecting high model fidelity and minimal phase lag in reproducing non-burst EMG noise behavior.

A reduced-length segment (399 samples, $F_s = 2000$ Hz) from the *in-move non-burst* phase of Dataset 4 (EMG1, Trial 05) was modelled using an ARIMA (1, 0, 3) process. The corresponding equation is expressed as:

$$x_t = \varepsilon_t + (0.5527B)x_t + (1.0526B + 0.9775B^2 + 0.5932B^3)\varepsilon_t \quad (7)$$

The Model generated an $AIC = -8653.56$ and an $AICc = -8633.76$, and the residual variance = 2.18×10^{-11}). The low differencing order ($d=0$) means the segment was intrinsically in motion and it is in line with inherently short and noise-limited signal. The shorter window length in comparison to the previous experiments also lowered the complexity of time but maintained an adequate dynamic range such that allowed AR and MA estimation to be done in a stable manner.

The fitted curve (Figure 31) exhibits a high rate of tracking the original signal, both in following high-frequency micro-oscillations and low-application similes of motion-steady noise. This result of the residual ACF (Figure 32) also shows that all of the autocorrelations are within the 95 percent confidence limits, demonstrating the whiteness and independence of the residuals.

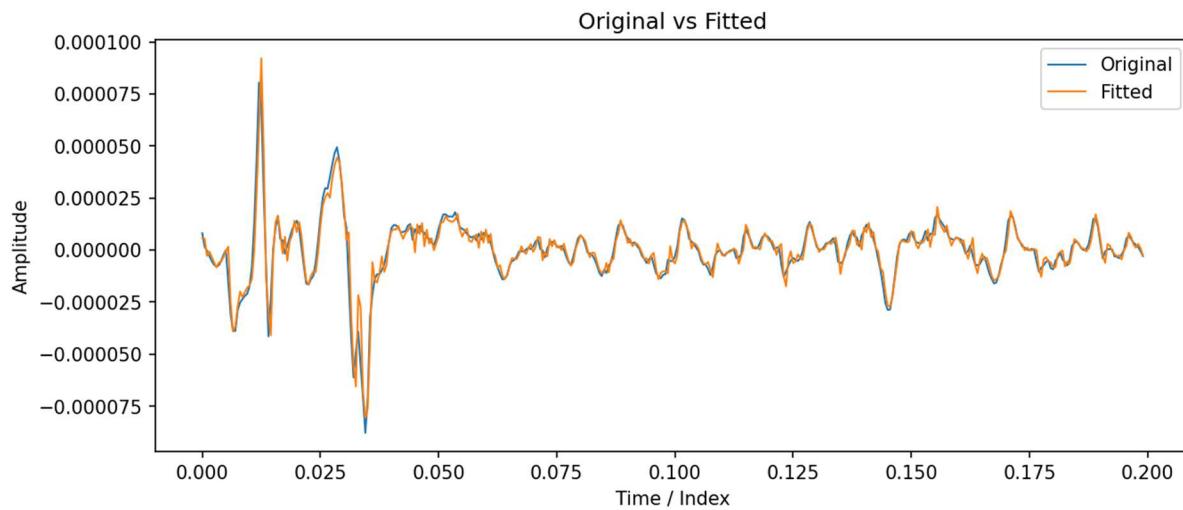


Figure 31. Original versus fitted ARIMA (1, 0, 3) model for Dataset 4 (EMG1, Trial 05)

The fitted sequence is very similar to the signal that was measured, which indicates strong short-term prediction and negligible amplitude bias.

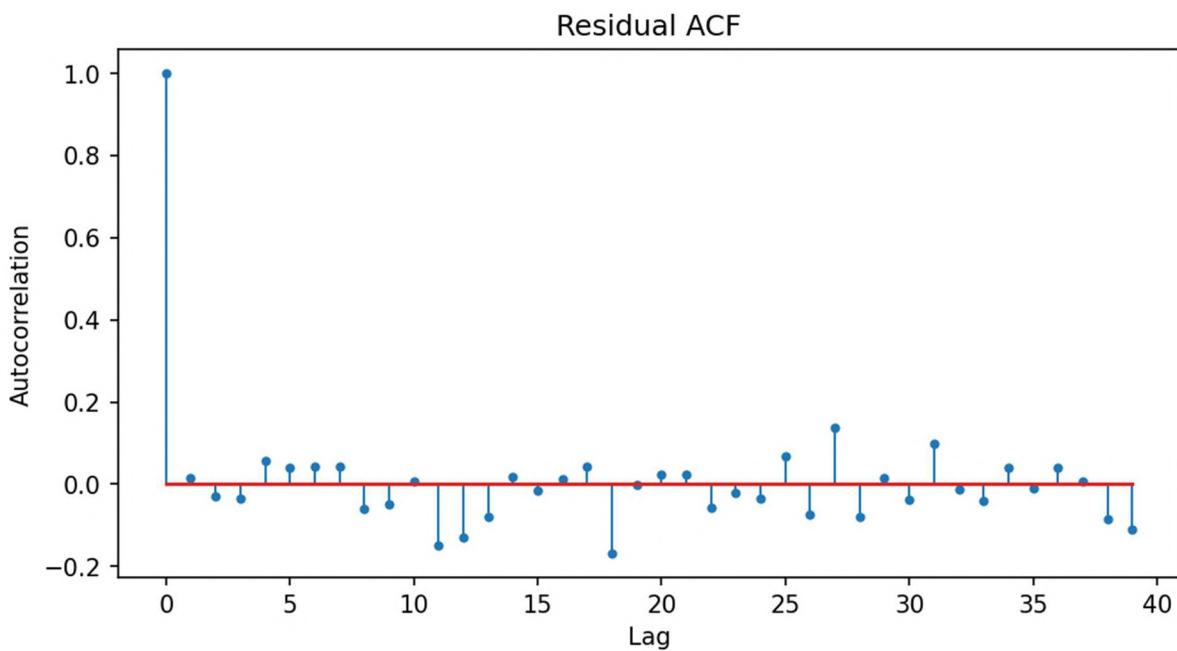


Figure 32. Residual autocorrelation (ACF) for the ARIMA (1, 0, 3) model applied to Dataset 4 (EMG1, Trial 05) (Dataset IV)

There is no substantial autocorrelation between the residuals and the lags, which confirms that the model is adequate and noise independent.

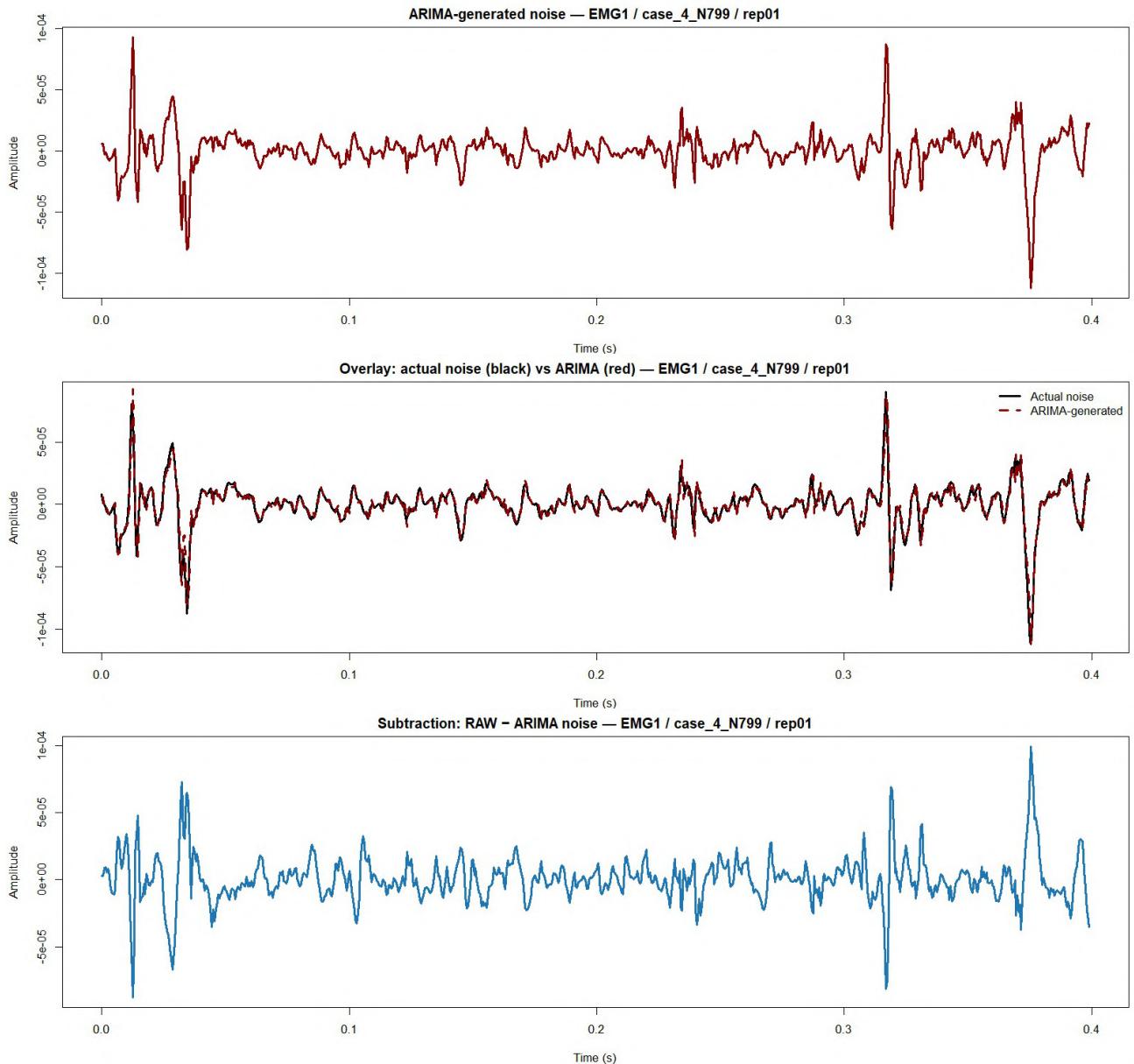


Figure 33. ARIMA-based noise modeling and subtraction (Dataset IV)

Figure 33 illustrates the application of the ARIMA-generated noise subtraction method on Dataset IV. The red waveform located in the top panel represents the noise pattern as generated by the fitted ARIMA model. This modeled noise (red, dashed) is compared to the actual measured noise (black) on the middle panel, and it is possible to see that the ARIMA output is very close to the actual noise waveform in terms of its amplitude and timing. The lower panel shows the variations between the raw EMG signal and between the noise produced by the ARIMA (RAW - ARIMA). This subtraction indicates the contents of the EMG activity that are retained when the modeled noise is subtracted, and it shows how the ARIMA equation can be

applied in order to suppress the background interference and retain the main features in the signal.

4.7 Key Findings

The separation of unwanted noise of actual muscle activity in the electromyography records was successfully achieved in the analysis. The noise sequences were in most instances stable and they did not need a lot of correction to stabilise them. The overall patterns of noise could be effectively kept by simple ARIMA models, and in the case of ARIMA models the residuals looked like random white noise. On the majority of recordings, the steady behavior was observed with minimum drift ($d \leq 1$), but some short or highly variable samples were on the borderline.

There were numerous cases when the low-order ARIMA models have worked perfectly, yet there are cases in which the full provision of the residual whiteness condition (Ljung-Box $p < 0.05$) has not been satisfied, so the structure has not been entirely explained. In some of the recordings, periodic interference close to 50 or 100 Hz still remained and it would be possible that a model with seasonal or sinusoidal terms such as SARIMA would be closer to existence in those instances. There was no major difference in the results obtained through each dataset, however, some of the channels or gestures were somewhat unstable or unpredictable. Table 9 shows root mean square error of ARIMA residuals across all datasets.

Table 9. ARIMA Root Mean Square Error of Residuals across all four datasets for rest gesture

Dataset-I			Dataset -II		Dataset -III		Dataset -IV	
Channels	Samples	RMSE_residuals	Samples	RMSE_residuals	Samples	RMSE_residuals	Samples	RMSE_residuals
Ch1	351	0.0057691	375	0.000149	528	4.93E-06	550	5.0165E-06
Ch2	351	0.0063323	375	9.96E-06	528	6.652E-06	550	1.0462E-05
Ch3	351	0.0189854	375	1.38E-06	528	9.834E-06	550	3.1101E-06
Ch4	351	0.0119574	375	1.16E-06	528	5.269E-06	550	2.1122E-06
Ch5	351	0.0063517	N/A	N/A	528	4.659E-06	550	3.7545E-06
Ch6	351	0.0081841	N/A	N/A	528	5.105E-06	550	7.1262E-06
Ch7	351	0.0257847	N/A	N/A	528	5.038E-06	550	2.1451E-06
Ch8	351	0.0155571	N/A	N/A	528	4.111E-06	550	2.6853E-06
Ch9	N/A	N/A	N/A	N/A	N/A	N/A	550	3.3291E-06
Ch10	N/A	N/A	N/A	N/A	N/A	N/A	550	1.1656E-05
Ch11	N/A	N/A	N/A	N/A	N/A	N/A	550	2.2795E-06
Ch12	N/A	N/A	N/A	N/A	N/A	N/A	550	3.3678E-06

5 CONCLUSION

This thesis was based on questioning whether the noise dynamics in electromyography (EMG) can be well explained and reproduced using autoregressive integrated moving average (ARIMA) models. Since clean (noise free) datasets are unusual, the project started by decoupling noise directly with unfiltered EMG signals and then trying to model the noise. The primary concept was easy: as soon as the power-line hum was eliminated and muscle activity had been filtered out, the residual signal would have steady behavior on short-period basis to be orchestrated by ARIMA models.

An entire functional workflow was developed and evaluated using various datasets containing a number of EMG channels. The first method of organizing recordings was based on subject, gesture and repetition. The 50-Hz power line noise and its harmonic components were eliminated by sinusoid fitting rather than rigid filters, as this technique did not disarm the natural frequency structure of the muscle signal. Automatic identification of noise portions was then done by a sliding RMS energy envelope and a threshold based on data and short silent blocks were subsequently combined to proceed with additional analysis. All the noise sequences extracted were verified using frequency and time domain tests to ensure that the noise sequences acted more like random noise and less like residual muscle activity.

The modeling phase was looking at the quality of the low-order ARIMA models to explain these noise sequences. In both channels, several cases were generated by making successively shorter samples (such as full length, half, and quarter). Signal stability or stationarity was tested and difference when there was slow drift was applied. The selection of models was done with care in order to achieve reconciliation between accuracy and simplicity and performance of the models was verified by standard residual tests to ensure the remaining errors were random white noises. The modeled noise in a few instances was subtracted in mixed EMG-plus-noise recordings to indicate signal potentially being improved.

A prevailing trend was observed. Simple ARMA models often adequately modeled the noise, and a few demanded a single step of differencing in order to get smooth results. Short windows usually provided more consistent fits, which is consistent with the notion of EMG noise being more consistent in the short-term. Essentially constant channels caused such edge cases to act as pure random signals, and slow drifts required differencing to cause the model to hold.

There were certain limitations anticipated. It was necessary to check model accuracy against the indirect scales such as variance or spectrum changes and not a consistent clean definition of the EMG. It was specifically intended to be used in short, locally stable segments and when

used in long-term, non-stationary artefacts would require regular refitting of its models. The light length of windows may increase variability and the chances of overfitting as well. Channels with complicated non-sinusoidal interference might need further or numerous-channel modeling in future.

Despite these difficulties, the findings prove that ARIMA models can be a useful and interpretable tool of understanding, simulating and subtracting noises in EMG signals-assuming the noisy impurity is correctly dissociated, the model undergoes stationarity tests and the pattern of its remains are thoroughly examined. The persistence in the observed improvement of datasets indicates that short-window modeling is reliable in this case. Based on the available evidence, it is the first study, where the EMG noise component is formally modeled instead of simply filtered. The study shows that noise behavior can be learned and reproduced statistically, by considering noise as a structured phenomenon, and that principled subtraction of mixed EMG recordings is possible, which allows the study to be considered as principled.

Advantages:

- Step-by-step workflow with measurable checks at each stage (hum removal, noise extraction, modeling).
- Targeted 50/60 Hz noise removal that avoids damage to neighboring frequency bands.
- Real noise data that can be reused for simulation, algorithm testing, and quality monitoring.
- Short-window modeling strategy that increases signal stability and improves accuracy.
- Simple, interpretable ARIMA fits with transparent validation through residual checks.
- Low computational cost and easy adaptation per channel or dataset.
- Consistent performance across datasets with minimal parameter adjustments.

Challenges and Future Work:

Dissimilarity in the format of data sets like sampling rates, channels and time can occasionally lead to gaps in results or wobbly outcomes, which constrained model validation. Some of the recordings still demonstrated residual sidebands or baseline drift despite regression and smaller windows had to be used. A lack of a ground truth of clean EMG resulted in the fact that the clean EMG could be evaluated by indirect metrics but by using very short samples, one would have an overfit model unless the orders of the models were limited. Next, the basic noise baseline of individual subjects can be logged and documented and the mains signal logged to facilitate smoother adaptive noise reduction, metadata and thresholds can be standardized

across all datasets, and quality checks implemented automatically prior to model generation. The framework can further be extended to multi channel and adaptive model towards enhancing better accuracy and generalization. Collectively, these improvements would enhance the reliability of EMG noise modeling, cross dataset generalizability, and the application of EMG noise modeling in the research and clinical environments.

REFERENCES

- [1] Roberto. Merletti and Dario. Farina, *Surface electromyography: physiology, engineering, and applications*. IEEE Press, 2016.
- [2] Roberto. Merletti and Philip. Parker, *Electromyography: physiology, engineering, and noninvasive applications*. Wiley-Interscience, 2005.
- [3] M. B. I. Reaz, M. S. Hussain, and F. Mohd-Yasin, “Techniques of EMG signal analysis: detection, processing, classification and applications,” *Biol Proced Online*, vol. 8, no. 1, pp. 11–35, Dec. 2006, doi: 10.1251/bpo115.
- [4] B. Widrow *et al.*, “Adaptive noise cancelling: Principles and applications,” *Proceedings of the IEEE*, vol. 63, no. 12, pp. 1692–1716, 1975, doi: 10.1109/PROC.1975.10036.
- [5] C. J. De Luca, A. Adam, R. Wotiz, L. D. Gilmore, and S. H. Nawab, “Decomposition of Surface EMG Signals,” *J Neurophysiol*, vol. 96, no. 3, pp. 1646–1657, Sep. 2006, doi: 10.1152/jn.00009.2006.
- [6] R. M. . Rangayyan, *Biomedical signal analysis*. John Wiley & Sons, Inc., 2015.
- [7] P. J. Brockwell and R. A. Davis, *Introduction to Time Series and Forecasting*. Cham: Springer International Publishing, 2016. doi: 10.1007/978-3-319-29854-2.
- [8] O. Paiss and G. F. Inbar, “Autoregressive Modeling of Surface EMG and Its Spectrum with Application to Fatigue,” *IEEE Trans Biomed Eng*, vol. BME-34, no. 10, pp. 761–770, Oct. 1987, doi: 10.1109/TBME.1987.325918.
- [9] A. de Cheveigné, “ZapLine: A simple and effective method to remove power line artefacts,” *Neuroimage*, vol. 207, p. 116356, 2020, doi: <https://doi.org/10.1016/j.neuroimage.2019.116356>.
- [10] D. Esposito, J. Centracchio, P. Bifulco, and E. Andreozzi, “A smart approach to EMG envelope extraction and powerful denoising for human–machine interfaces,” *Sci Rep*, vol. 13, no. 1, p. 7768, May 2023, doi: 10.1038/s41598-023-33319-4.
- [11] P. J. Huber, *Robust Statistics*. Wiley, 1981. doi: 10.1002/0471725250.

- [12] P. Welch, “The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms,” *IEEE Transactions on Audio and Electroacoustics*, vol. 15, no. 2, pp. 70–73, Jun. 1967, doi: 10.1109/TAU.1967.1161901.
- [13] F. J. Harris, “On the use of windows for harmonic analysis with the discrete Fourier transform,” *Proceedings of the IEEE*, vol. 66, no. 1, pp. 51–83, 1978, doi: 10.1109/PROC.1978.10837.
- [14] A. V. . Oppenheim and R. W. . Schafer, *Discrete-time signal processing*. Pearson, 2010.
- [15] D. A. Dickey and W. A. Fuller, “Distribution of the Estimators for Autoregressive Time Series With a Unit Root,” *J Am Stat Assoc*, vol. 74, no. 366, pp. 427–431, 1979, doi: 10.2307/2286348.
- [16] G. E. P. . Box, G. M. . Jenkins, G. C. . Reinsel, and G. M. . Ljung, *Time series analysis : forecasting and control*. John Wiley & Sons, Inc., 2016.
- [17] D. Kwiatkowski, P. C. B. Phillips, P. Schmidt, and Y. Shin, “Testing the null hypothesis of stationarity against the alternative of a unit root,” *J Econom*, vol. 54, no. 1–3, pp. 159–178, Oct. 1992, doi: 10.1016/0304-4076(92)90104-Y.
- [18] P. C. B. Phillips and P. Perron, “Testing for a Unit Root in Time Series Regression,” *Biometrika*, vol. 75, no. 2, p. 335, Jun. 1988, doi: 10.2307/2336182.
- [19] C. M. HURVICH and C.-L. TSAI, “Regression and time series model selection in small samples,” *Biometrika*, vol. 76, no. 2, pp. 297–307, 1989, doi: 10.1093/biomet/76.2.297.
- [20] R. J. Hyndman and Y. Khandakar, “Automatic Time Series Forecasting: The **forecast** Package for R,” *J Stat Softw*, vol. 27, no. 3, 2008, doi: 10.18637/jss.v027.i03.
- [21] A. Hyvärinen and E. Oja, “Independent component analysis: algorithms and applications,” *Neural Networks*, vol. 13, no. 4–5, pp. 411–430, Jun. 2000, doi: 10.1016/S0893-6080(00)00026-5.
- [22] Z. WU and N. E. HUANG, “ENSEMBLE EMPIRICAL MODE DECOMPOSITION: A NOISE-ASSISTED DATA ANALYSIS METHOD,” *Adv Adapt Data Anal*, vol. 01, no. 01, pp. 1–41, Jan. 2009, doi: 10.1142/S1793536909000047.
- [23] N. E. Huang *et al.*, “The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis,” *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 454, no. 1971, pp. 903–995, Mar. 1998, doi: 10.1098/rspa.1998.0193.
- [24] D. B. Percival and A. T. Walden, *Wavelet Methods for Time Series Analysis*. Cambridge: Cambridge University Press, 2000. doi: 10.1017/CBO9780511841040.

- [25] D. L. Donoho and I. M. Johnstone, “Ideal Spatial Adaptation by Wavelet Shrinkage,” *Biometrika*, vol. 81, no. 3, p. 425, Aug. 1994, doi: 10.2307/2337118.
- [26] G. M. Ljung and G. E. P. Box, “On a Measure of Lack of Fit in Time Series Models,” *Biometrika*, vol. 65, no. 2, p. 297, Aug. 1978, doi: 10.2307/2335207.
- [27] “FORS-EMG: A Novel sEMG Dataset.” Accessed: Sep. 23, 2025. [Online]. Available: <https://www.kaggle.com/datasets/ummerummamchaity/fors-emg-a-novel-semg-dataset>
- [28] H. J. Hermens *et al.*, “European recommendations for surface electromyography: Results of the SENIAM Project,” 1999. [Online]. Available: <https://api.semanticscholar.org/CorpusID:114598925>
- [29] M. A. Ozdemir, D. H. Kisa, O. Guren, and A. Akan, “Dataset for multi-channel surface electromyography (sEMG) signals of hand gestures,” *Data Brief*, vol. 41, p. 107921, Apr. 2022, doi: 10.1016/j.dib.2022.107921.
- [30] S. Lobov, N. Krilova, I. Kastalskiy, V. Kazantsev, and V. Makarov, “Latent Factors Limiting the Performance of sEMG-Interfaces,” *Sensors*, vol. 18, no. 4, p. 1122, Apr. 2018, doi: 10.3390/s18041122.
- [31] A. Krasoulis, I. Kyranou, M. S. Erden, K. Nazarpour, and S. Vijayakumar, “Improved prosthetic hand control with concurrent use of myoelectric and inertial measurements,” *J Neuroeng Rehabil*, vol. 14, no. 1, p. 71, Dec. 2017, doi: 10.1186/s12984-017-0284-4.

APPENDIX A : DATASET SUMMARIES**D3****Table 10. Dataset III Detailed acquisition summary**

Field	Summary
Source & device	EMG Pattern Database; MYO Thalmic bracelet on forearm; data streamed via Bluetooth to PC.
Subjects	36 participants.
Channels	8 surface EMG sensors, equally spaced around the forearm.
File structure	Plain-text tables with 10 columns: Time (ms), Channels 1–8, Class label.
Labels (Class)	0 = unmarked; 1 = rest; 2 = fist; 3 = wrist flexion; 4 = wrist extension; 5 = radial deviation; 6 = ulnar deviation; 7 = extended palm (optional).
Trial protocol	Two series per subject; each series comprises six (or seven) static gestures, each held ~3 s with ~3 s pauses.
Time base	Time in milliseconds (column 1); sampling rate not explicitly stated in the README.
Use in this project	Per-channel segmentation by class label; extraction of rest/gesture epochs; noise verification and ARIMA modelling on the 8 EMG channels.

D4**Table 11. Dataset IV Detailed Acquisition Summary**

Field	Summary
Subject	Subject_2 (single participant instance used for results)
Acquisition device	Delsys Trigno IM: 12 double-differential sEMG channels with co-located 9-axis IMUs (accelerometer, gyroscope, magnetometer)
Sampling rate (EMG)	2000 Hz (project configuration)
Exercises covered	Exercise B (IDs 1–17) and Exercise C (IDs 18–40)
Gesture mapping	Exercise B examples include: Thumb up; Pointing index; Wrist flexion/extension; radial/ulnar deviation, etc. Exercise C covers grasp types (e.g., tripod, lateral, power/precision spheres) and functional actions (e.g., screwdriver, bottle opening, knife cut). Each gesture is stored in a folder with rep01...rep06 files.
Repetitions per gesture	6 repetitions per gesture (rep01–rep06) for both exercises.
Channels used in analysis	12 EMG channels (per subject), with IMU streams available but not required for the ARIMA noise modeling in this chapter
Trial length	~5 s active + ~3 s rest per repetition (per dataset protocol)
Total trials for this subject	40 gestures × 6 reps = 240 trials (Exercise B: 17×6=102; Exercise C: 23×6=138). Folder counts confirm all expected gesture folders present with six files each.

APPENDIX B : ADDITIONAL RESULTS

ARIMA Models

D-I :

Table 12. ARIMA modelling results across progressive subsampling cases (DI-Ch1)

Case	N	Differencing d	ARIMA (p,d,q)	RMSE (resid.)	R ² (noise)	Ljung–Box p
case_1_N5626	5626	0	(2,0,3)	0.032295	0.375	1.52E-06
case_2_N2813	2813	0	(2,0,3)	0.029602	0.4	0.00484
case_3_N1406	1406	1	(5,1,0)	0.02606	0.251	6.60E-10
case_4_N703	703	0	(2,0,3)	0.007916	0.27	0.0288
case_5_N351	351	0	(3,0,2)	0.005769	0.31	0.00261

D-II

Table 13. ARIMA modeling across halving cases (Dataset II : Ch1)

Case	N	Differencing used (d)	ARIMA (p,d,q)	RMSE (resid.)	AICc
case_1_N12000	12,000	1	(5,1,0)	5.38 x10 ⁻⁶	-257080.72
case_2_N6000	6,000	1	(5,1,0)	4.57 x10 ⁻⁶	-130470.91
case_3_N3000	3,000	0	(1,0,0)	3.47 x10 ⁻⁶	-25,463.32
case_4_N1500	1,500	0	(0,0,5)	1.29 x10 ⁻⁶	-15,665.87
case_5_N750	750	0	(3,0,0)	1.43 x10 ⁻⁶	-11,140.81
case_6_N375	375	0	(3,0,0)	1.49 x10 ⁻⁶	-5,534.73

D-III**Table 14. ARIMA modeling results (Dataset III : Ch1)**

Trial	Samples	Model (ARIMA p,d,q)	AR coeffs	MA coeffs	AIC	σ^2 (innovation)	KPSS p_value	Chosen d
T01	1171	(0, 1, 1)	N/A	-0.123	-25119.494	2.77×10^{-11}	0.013	1
T02	585	(1, 0, 1)	0.93	-0.130	-12681.620	2.63×10^{-11}	0.01	0
T03	500	(1, 0, 1)	0.933	-0.112	-10,878.369	2.63×10^{-11}	0.01	0

D-IV**Table 15. ARIMA modeling results (Dataset IV : Ch1)**

Trial	N (samples)	F _s (Hz)	Model	AICc	σ^2 (innovation)	Ljung–Box lag	Ljung–Box p
1	6,393	2000	ARIMA(2,0,3)	-136,639.356	3.05×10^{-11}	20	0
2	3,196	2000	ARIMA(2,0,3)	-67,036.470	4.54×10^{-11}	20	3.23×10^{-10}
3	1,598	2000	ARIMA(2,0,3)	-33,412.009	4.83×10^{-11}	20	2.97×10^{-3}
4	799	2000	ARIMA(1,0,3)	-17,218.806	2.53×10^{-11}	20	8.07×10^{-3}
5	399	2000	ARIMA(1,0,3)	-8,653.556	2.18×10^{-11}	20	1.15×10^{-3}

APPENDIX C : CODE REFERENCES

Segmentation

```

req <- c("data.table", "R.matlab", "tools", "stringr")
for (p in req) if (!requireNamespace(p, quietly = TRUE)) install.packages(p)
library(data.table); library(R.matlab); library(stringr)

fors_root <- "C:/Users/nabee/New folder/IAP/archive(2)/FORS-EMG Dataset/FORS-EMG
Dataset/FORS-EMG"
out_root <- "C:/Users/nabee/New folder/IAP/segments_out_fors"
dir.create(out_root, showWarnings = FALSE, recursive = TRUE)

Fs      <- 985L
dur_sec <- 8L
n_ch_exp <- 8L

gesture_map <- c(
  "Thumb_UP"        = "TU",
  "Index"           = "IDX",
  "Right_Angle"     = "RA",
  "Peace"            = "PCE",
  "Index_Little"    = "IL",
  "Thumb_Little"    = "TL",
  "Hand_Close"      = "HC",
  "Hand_Open"        = "HO",
  "Wrist_Extension" = "WE",
  "Wrist_Flexion"   = "WF",
  "Ulner_Deviation" = "UD",
  "Radial_Deviation"= "RD"
)

orient_norm <- function(s) {
  s <- tolower(s)
  if (grepl("sup", s)) return("supination")
  if (grepl("pro", s)) return("pronation")
  if (grepl("rest|neutral", s)) return("rest")
  return(s)
}

load_mat_matrix <- function(path) {
  m <- R.matlab::readMat(path)
  cand_names <- c("data", "SEMG", "emg", "X", "sig", "signal")
  for (nm in cand_names) {
    if (!is.null(m[[nm]]) && (is.numeric(m[[nm]]) || is.matrix(m[[nm]]))) {
      X <- m[[nm]]
      X <- if (is.vector(X)) matrix(X, ncol=1L) else as.matrix(X)
      storage.mode(X) <- "double"
      if (nrow(X) < ncol(X)) X <- t(X)
      return(X)
    }
  }
  nums <- m[vapply(m, function(z) is.numeric(z) || is.matrix(z), TRUE)]
  if (length(nums) == 0) stop("No numeric array in MAT: ", path)
  nums <- m[vapply(m, function(z) is.numeric(z) || is.matrix(z), TRUE)]
  if (length(nums) == 0) stop("No numeric array in MAT: ", path)
  sz <- vapply(nums, function(z) length(as.vector(z)), numeric(1))
  X <- as.matrix(nums[[which.max(sz)]])
  if (nrow(X) < ncol(X)) X <- t(X)
  storage.mode(X) <- "double"
  return(X)
  X <- as.matrix(nums[[which.max(sz)]])
  if (nrow(X) < ncol(X)) X <- t(X)
  storage.mode(X) <- "double"
  X
}

save_segment <- function(x, out_dir, g_code, trial_id, Fs, ori, subject_id, src_file)
{
  dir.create(out_dir, recursive = TRUE, showWarnings = FALSE)
  stem <- sprintf("multICH_%s_trial%s", g_code, trial_id)
  fwrite(as.data.table(x), file = file.path(out_dir, paste0(stem, ".csv")))
  saveRDS(x, file = file.path(out_dir, paste0(stem, ".rds")))
}

```

```

for (ch in seq_len(ncol(x))) {
  sc_stem <- sprintf("singleCH_%s_trial%s_CH%d", g_code, trial_id, ch)
  fwrite(data.table(x[,ch]), file = file.path(out_dir, paste0(sc_stem, ".csv")))
  saveRDS(x[,ch], file = file.path(out_dir, paste0(sc_stem, ".rds")))
}
meta <- list(
  subject      = subject_id,
  orientation   = ori,
  gesture       = g_code,
  trial         = trial_id,
  Fs            = Fs,
  duration_s    = nrow(x)/Fs,
  n_samples     = nrow(x),
  n_channels    = ncol(x),
  source_file   = src_file
)
saveRDS(meta, file = file.path(out_dir, sprintf("meta_%s_trial%s.rds", g_code, trial_id)))
}

parse_gesture_trial <- function(fname) {
  base <- tools::file_path_sans_ext(basename(fname))
  m <- regexec("^(.*)(\\d+)$", base)
  r <- regmatches(base, m)[[1]]
  if (length(r) == 3) {
    gest <- r[2]; tr <- r[3]
  } else {
    parts <- strsplit(base, "-|_trial|_TR|_|")[[1]]
    gest <- parts[1]; tr <- parts[length(parts)]
  }
  list(gesture=gest, trial=tr)
}

gesture_to_code <- function(gest_raw) {
  if (!is.na(gesture_map[gest_raw])) return(unname(gesture_map[gest_raw]))
  idx <- pmatch(tolower(gest_raw), tolower(names(gesture_map)))
  if (!is.na(idx)) return(unname(gesture_map[idx]))
  if (toupper(gest_raw) %in% gesture_map) return(toupper(gest_raw))
  toupper(substr(gsub("[^A-Za-z]", "", gest_raw), 1, 3))
}

subjects <- list.dirs(fors_root, recursive = FALSE, full.names = TRUE)
subjects <- subjects[dir.exists(subjects)]

cat("Found", length(subjects), "subject folders\n")

for (sdir in subjects) {
  subj_name <- basename(sdir)
  subj_idx <- suppressWarnings(as.integer(gsub("\\D", "", subj_name)))
  subj_tag <- if (!is.na(subj_idx)) sprintf("subject_%02d", subj_idx) else subj_name
  oris <- list.dirs(sdir, recursive = FALSE, full.names = TRUE)
  oris <- oris[dir.exists(oris)]
  for (odir in oris) {
    ori_tag <- orient_norm(basename(odir))
    mats <- list.files(odir, pattern="\\.mat$", full.names=TRUE)
    if (length(mats) == 0) next
    for (mf in mats) {
      info <- parse_gesture_trial(mf)
      g_code <- gesture_to_code(info$gesture)
      trial <- info$trial
      x <- load_mat_matrix(mf)
      if (ncol(x) != n_ch_exp) {
        warning(sprintf("Unexpected channel count (%d) in %s; continuing.", ncol(x), mf))
      }
      need_n <- Fs * dur_sec
      if (nrow(x) > need_n) x <- x[seq_len(need_n), , drop=FALSE]
      if (nrow(x) < need_n) {
        pad <- matrix(0, nrow=need_n - nrow(x), ncol=ncol(x))
        x <- rbind(x, pad)
      }
      out_dir <- file.path(out_root, subj_tag, ori_tag, g_code)
      save_segment(x, out_dir, g_code, trial, Fs, ori_tag, subj_tag, src_file = mf)
      cat(sprintf("Saved %s | %s | %s trial %s -> %s\n", subj_tag, ori_tag, g_code, trial, out_dir))
    }
  }
}

```

```

}

cat("\nDone. Segments written under:\n", out_root, "\n")

```

Noise Acquisition

```

req <- c("data.table", "zoo", "tools")
for (p in req) if (!requireNamespace(p, quietly = TRUE)) install.packages(p)
library(data.table); library(zoo); library(tools)

req2 <- c("e1071", "tseries", "forecast", "ggplot2")
for (p in req2) if (!requireNamespace(p, quietly = TRUE)) install.packages(p)
library(e1071)
library(tseries)
library(forecast)
library(ggplot2)

segments_root <- "C:/Users/nabee/New folder/IAP/segments_out_fors"
noise_root <- "C:/Users/nabee/New folder/IAP/noise_out_fors_gated"
dir.create(noise_root, recursive = TRUE, showWarnings = FALSE)

Fs          <- 985L
f0          <- 50
harmonics   <- 1:3
win_env_s    <- 0.10
thr_k        <- 1.5
min_quiet_s  <- 0.20
max_gap_s    <- 0.05
SUBJECTS_MAX <- NA
MIN_VAR      <- 1e-12

as_time_by_channel <- function(x) {
  x <- as.matrix(x); storage.mode(x) <- "double"
  if (nrow(x) < ncol(x)) x <- t(x)
  x
}

rm_dc <- function(x) sweep(x, 2, colMeans(x, na.rm=TRUE), "-")

pli_design <- function(N, Fs, f0, harmonics) {
  t <- (0:(N-1))/Fs
  cols <- lapply(harmonics, function(h) cbind(sin(2*pi*h*f0*t), cos(2*pi*h*f0*t)))
  do.call(cbind, cols)
}

estimate_pli_vec <- function(x, Fs, f0=50, harmonics=1:3) {
  x <- as.numeric(x); N <- length(x)
  D <- pli_design(N, Fs, f0, harmonics)
  x0 <- x - mean(x)
  beta <- tryCatch(qr.solve(D, x0), error=function(e) rep(0, ncol(D)))
  as.vector(D %*% beta)
}

rms_envelope <- function(x, Fs, win_s=0.10) {
  w <- max(1L, round(win_s * Fs))
  m <- zoo::rollapply(x^2, width=w, FUN=mean, align="center", fill="extend")
  sqrt(as.numeric(m))
}

quiet_mask_from_env <- function(env, Fs, k=1.5, min_quiet_s=0.2, max_gap_s=0.05) {
  if (!is.numeric(env) || !length(env)) return(rep(FALSE, length(env)))
  thr <- median(env, na.rm=TRUE) + k * mad(env, constant=1.4826, na.rm=TRUE)
  raw <- env < thr
  raw[is.na(raw)] <- FALSE
  max_gap <- max(1L, round(max_gap_s * Fs))
  r <- rle(raw)
  if (length(r$values) > 2 && max_gap > 0) {
    for (i in seq_along(r$values)) {
      if (!r$values[i] && r$lengths[i] <= max_gap) {
        left_quiet <- if (i > 1) r$values[i-1] else FALSE
        right_quiet <- if (i < length(r$values)) r$values[i+1] else FALSE
        if (left_quiet && right_quiet) r$values[i] <- TRUE
      }
    }
  }
  mask <- inverse.rle(r)
}

```

```

min_len <- max(1L, round(min_quiet_s * Fs))
r2 <- rle(mask)
for (i in seq_along(r2$values)) {
  if (r2$values[i] && r2$lengths[i] < min_len) r2$values[i] <- FALSE
}
inverse.rle(r2)

spec_flatness <- function(x) {
  sp <- spec.pgram(x, taper=0, plot=FALSE, fast=TRUE)
  p <- sp$spec + .Machine$double.eps
  as.numeric(exp(mean(log(p))) / mean(p))
}

rms <- function(x) sqrt(mean(x^2, na.rm=TRUE))

psd_welch <- function(x, Fs) {
  sp <- spec.pgram(x, taper=0, plot=FALSE, fast=TRUE)
  list(f = sp$freq * Fs/2, p = sp$spec + .Machine$double.eps)
}

bandpower <- function(f, p, f1, f2) {
  idx <- which(f >= f1 & f <= f2)
  if (length(idx) < 2) return(0)
  sum(p[idx])
}

line_hum_score <- function(x, Fs, f0=50, harmonics=1:3, bw=1.5) {
  ps <- psd_welch(x, Fs); ptot <- sum(ps$p); if (ptot == 0) return(0)
  num <- 0
  for (h in harmonics) {
    idx <- which(ps$f >= (h*f0 - bw) & ps$f <= (h*f0 + bw))
    num <- num + sum(ps$p[idx])
  }
  as.numeric(num/ptot)
}

lf_ratio <- function(x, Fs, lf_max=20) {
  ps <- psd_welch(x, Fs); ptot <- sum(ps$p); if (ptot==0) return(0)
  as.numeric(bandpower(ps$f, ps$p, 0, lf_max)/ptot)
}

hf_ratio <- function(x, Fs, hf_min=150) {
  ps <- psd_welch(x, Fs); ptot <- sum(ps$p); if (ptot==0) return(0)
  as.numeric(bandpower(ps$f, ps$p, hf_min, max(ps$f))/ptot)
}

spike_index <- function(x, T=6) {
  z <- (x - median(x)) / (mad(x, constant=1.4826) + 1e-12)
  mean(abs(z) > T, na.rm=TRUE)
}

cardiac_like_score <- function(x, Fs) {
  ps <- psd_welch(x, Fs)
  band <- bandpower(ps$f, ps$p, 5, 40) / (sum(ps$p) + 1e-12)
  ac <- acf(x, lag.max = floor(1.2*Fs), plot = FALSE)$acf[-1]
  lags_ms <- (1:length(ac))/Fs*1000
  peri <- max(abs(ac[lags_ms] >= 200 & lags_ms <= 1200)), na.rm=TRUE)
  as.numeric(0.7*band + 0.3*peri)
}

common_mode_score <- function(x) {
  if (ncol(x) < 2) return(0)
  R <- cor(x, use="pairwise.complete.obs")
  mean(R[upper.tri(R)], na.rm=TRUE)
}

stationarity_flag_noise <- function(x) {
  padf <- tryCatch(adf.test(x)$p.value, error=function(e) NA_real_)
  pkpss <- tryCatch(kpss.test(x, null="Level")$p.value, error=function(e) NA_real_)
  if (is.na(padf) || is.na(pkpss)) return("unknown")
  if (padf < 0.05 && pkpss > 0.05) "stationary" else "non/near-stationary"
}

tag_noise_channel <- function(x, Fs, f0=50, harmonics=1:3) {
  if (length(x) < 200) {
    return(list(tags="too_short", score=0.0,

```

```

feats=data.frame(sf=NA,kurtosis=NA,line_ratio=NA,lf_ratio=NA,
hf_ratio=NA,spikes=NA,cardiac_like=NA,stationarity=NA)))
}
x <- as.numeric(x); x <- x[is.finite(x)]
sf <- spec_flatness(x)
kur <- e1071::kurtosis(x, type=2)
lrat <- line_hum_score(x, Fs, f0, harmonics)
rlf <- lf_ratio(x, Fs)
rhf <- hf_ratio(x, Fs)
spk <- spike_index(x)
car <- cardiac_like_score(x, Fs)
stat <- stationarity_flag_noise(x)
tags <- c(); conf <- c()
if (lrat > 0.15) { tags <- c(tags, "PowerLine");}
conf <- c(conf, min(1, (lrat-0.15)/0.15)) } { tags <- c(tags, "Baseline/Motion");}
if (rlf > 0.25 && sf < 0.6) { tags <- c(tags, "Transients/Pops");}
if (kur > 6 || spk > 0.002) { tags <- c(tags, "Cardiac-like Leakage");}
conf <- c(conf, min(1, (max(kur-6,0)/4 + min(spk/0.01,1))/2)) }
if (car > 0.20) { tags <- c(tags, "Residual-EMG");}
conf <- c(conf, min(1, (car-0.20)/0.20)) }
if (sf > 0.75 && rlf < 0.15 && lrat < 0.10) { tags <- c(tags, "Broadband-Instrumentation");}
conf <- c(conf, min(1, (sf-0.75)/0.15)) }
if (rhf > 0.25 && sf < 0.8) { tags <- c(tags, "Uncertain"); conf <- 0.2 }
feats <- data.frame(sf=sf, kurtosis=kur, line_ratio=lrat, lf_ratio=rlf,
hf_ratio=rhf,
spikes=spk, cardiac_like=car, stationarity=stat)
list(tags=paste(tags, collapse="|"), score=round(mean(conf),3), feats=feats)
}

verify_noise_trial <- function(x_noise, Fs, f0=50, harmonics=1:3) {
  X_noise <- as.matrix(x_noise); storage.mode(X_noise) <- "double"
  if (nrow(X_noise) == 0) {
    return(list(table=data.frame(channel=character(), tag=character(),
conf=numeric(), sf=numeric(), kurtosis=numeric(),
line_ratio=numeric(), lf_ratio=numeric(), hf_ratio=numeric(),
spikes=numeric(), cardiac_like=numeric(), stationarity=character(),
common_mode=numeric(), stringsAsFactors=FALSE),
common_mode=NA_real_))
  }
  C <- ncol(X_noise)
  cm <- common_mode_score(X_noise)
  rows <- vector("list", C)
  for (ch in seq_len(C)) {
    res <- tag_noise_channel(X_noise[,ch], Fs, f0, harmonics)
    rows[[ch]] <- data.frame(
      channel = paste0("CH", ch),
      tag = res$tags,
      conf = res$score,
      sf = res$feats$sf,
      kurtosis= res$feats$kurtosis,
      line_ratio = res$feats$line_ratio,
      lf_ratio = res$feats$lf_ratio,
      hf_ratio = res$feats$hf_ratio,
      spikes = res$feats$spikes,
      cardiac_like = res$feats$cardiac_like,
      stationarity = res$feats$stationarity,
      common_mode = cm,
      stringsAsFactors = FALSE
    )
  }
  list(table = do.call(rbind, rows), common_mode = cm)
}

plot_noise_psd <- function(x_noise, Fs, title) {
  df_all <- data.frame()
  for (ch in seq_len(ncol(x_noise))) {
    ps <- psd_welch(x_noise[,ch], Fs)
    df <- data.frame(f=ps$f, p=ps$p, ch=paste0("CH",ch))
    df_all <- rbind(df_all, df)
  }
}

```

```

ggplot(df_all, aes(f, 10*log10(p), color=ch)) + geom_line() +
  labs(title=title, x="Frequency (Hz)", y="PSD (dB)") + theme_minimal()
}

process_one_multich <- function(x, Fs, f0, harmonics,
                                 win_env_s, thr_k, min_quiet_s, max_gap_s) {
  x <- as_time_by_channel(x)
  if (var(as.vector(x), na.rm=TRUE) < MIN_VAR) return(NULL)
  x0 <- rm_dc(x)
  c <- ncol(x0)
  n_pli <- sapply(1:c, function(ci) estimate_pli_vec(x0[,ci], Fs, f0, harmonics))
  x1 <- x0 - n_pli
  env_mat <- sapply(1:c, function(ci) rms_envelope(x1[,ci], Fs, win_env_s))
  qmask_ch <- sapply(1:c, function(ci) {
    env <- env_mat[,ci]
    quiet_mask_from_env(env,           Fs,           k=thr_k,           min_quiet_s=min_quiet_s,
    max_gap_s=max_gap_s)
  })
  quiet_all <- apply(qmask_ch, 1, all)
  x_masked <- x1
  x_masked[!quiet_all, ] <- 0
  idx <- which(quiet_all)
  x_noise_cat <- if (length(idx)) x1[idx, , drop=FALSE] else matrix(0, nrow=0,
  ncol=c)
  diag <- data.frame(
    channel = paste0("CH", 1:c),
    quiet_frac = mean(quiet_all),
    rms_raw = apply(x, 2, rms),
    rms_afterPL = apply(x1, 2, rms),
    rms_noiseOR = if (nrow(x_noise_cat)) apply(x_noise_cat, 2, rms) else
    rep(NA_real_, c),
    flat_raw = apply(x, 2, spec_flatness),
    flat_afterPL = apply(x1, 2, spec_flatness),
    flat_noise = if (nrow(x_noise_cat)) apply(x_noise_cat, 2, spec_flatness) else
    rep(NA_real_, c),
    stringsAsFactors = FALSE
  )
  list(mask = quiet_all, env = env_mat, n_pli = n_pli,
       X_masked = X_masked, X_noise_cat = X_noise_cat, diag = diag)
}

save_noise_outputs <- function(res, out_dir, stem) {
  dir.create(out_dir, recursive = TRUE, showWarnings = FALSE)
  fwrite(as.data.table(res$X_masked), file.path(out_dir, paste0(stem,
  "_NOISE_MASKED.csv")))
  if (nrow(res$X_noise_cat) > 0) {
    fwrite(as.data.table(res$X_noise_cat), file.path(out_dir, paste0(stem,
  "_NOISE_CONCAT.csv")))
  } else {
    fwrite(data.table(), file.path(out_dir, paste0(stem, "_NOISE_CONCAT.csv")))
  }
  fwrite(as.data.table(res$n_pli), file.path(out_dir, paste0(stem, "_PLI.csv")))
  fwrite(data.table(quiet = as.integer(res$mask)),
        file.path(out_dir, paste0(stem, "_QUIET_MASK.csv")))
  fwrite(as.data.table(res$diag), file.path(out_dir, paste0(stem, "_DIAG.csv")))
}

subjects <- list.dirs(segments_root, recursive = FALSE, full.names = TRUE)
subjects <- subjects[dir.exists(subjects)]
if (!is.na(SUBJECTS_MAX)) subjects <- head(subjects, SUBJECTS_MAX)

manifest <- list()

for (sdir in subjects) {
  subj_tag <- basename(sdir)
  oris <- list.dirs(sdir, recursive = FALSE, full.names = TRUE)
  for (odir in oris) {
    ori_tag <- basename(odir)
    gdirs <- list.dirs(odir, recursive = FALSE, full.names = TRUE)
    for (gdir in gdirs) {
      gest_tag <- basename(gdir)
      files <- list.files(gdir, pattern="^multich_.*\\.csv$", full.names = TRUE)
      if (length(files) == 0) next
      for (f in files) {
        tmp <- data.table:::fread(f)
        X <- data.matrix(tmp); storage.mode(X) <- "double"
        X <- as_time_by_channel(X)
      }
    }
  }
}

```

```

    if (var(as.vector(x), na.rm=TRUE) < MIN_VAR) { warning("Near-constant: ", f); next }
    res <- process_one_multich(x, Fs, f0, harmonics, win_env_s, thr_k,
min_quiet_s, max_gap_s)
    if (is.null(res)) { warning("Processing returned NULL (near-constant?): ", f); next }
    rel_stem <- file_path_sans_ext(basename(f))
    out_dir <- file.path(noise_root,
                           basename(dirname(dirname(dirname(f)))),
                           basename(dirname(dirname(f))),
                           basename(dirname(f)))
    dir.create(out_dir, recursive = TRUE, showWarnings = FALSE)
    noise_to_check <- if (nrow(res$x_noise_cat) > 0) res$x_noise_cat else
res$x_masked
    vr <- verify_noise_trial(noise_to_check, Fs=F0, harmonics=harmonics)
    data.table::fwrite(vr$table, file.path(out_dir, paste0(rel_stem,
"_NOISE_VERIFY.csv")))
    png(file.path(out_dir, paste0(rel_stem, "_NOISE_PSD.png")), width=1400,
height=900)
    print(plot_noise_psd(noise_to_check, Fs=F0, title=paste0(rel_stem, " - noise
PSD")))
    dev.off()
    save_noise_outputs(res, out_dir, stem = rel_stem)
    trial_id <- sub("^.*_trial", "", rel_stem)
    manifest[[length(manifest)+1]] <- data.frame(
      subject = basename(dirname(dirname(f))),
      orientation = basename(dirname(dirname(f))),
      gesture = basename(dirname(f)),
      trial = trial_id,
      in_file = f,
      out_masked = file.path(out_dir, paste0(rel_stem, "_NOISE_MASKED.csv")),
      out_concat = file.path(out_dir, paste0(rel_stem, "_NOISE_CONCAT.csv")),
      verify_csv = file.path(out_dir, paste0(rel_stem, "_NOISE_VERIFY.csv")),
      psd_png = file.path(out_dir, paste0(rel_stem, "_NOISE_PSD.png")),
      stringsAsFactors = FALSE
    )
    cat(sprintf("OK (gated+verified): %s / %s / %s / %s\n",
                basename(dirname(dirname(f))),
                basename(dirname(dirname(f))),
                basename(dirname(f)),
                basename(f)))
  }
}
}

if (length(manifest)) {
  mf <- data.table::rbindlist(manifest, use.names = TRUE, fill = TRUE)
  fwrite(mf, file.path(noise_root, "manifest_noise_fors_gated.csv"))
  cat("\nsaved manifest to: ", file.path(noise_root,
"manifest_noise_fors_gated.csv"), "\n")
} else {
  cat("\nNo files processed. Check paths.\n")
}

```

ARIMA Modeling

```

req <- c("data.table", "forecast", "tseries", "urca", "ggplot2", "gridExtra", "tools")
for (p in req) if (!requireNamespace(p, quietly = TRUE)) install.packages(p)
library(data.table); library(forecast); library(tseries); library(urca)
library(ggplot2); library(gridExtra); library(tools)

noise_root <- "C:/Users/nabee/New folder/IAP/noise_out_fors"
arima_root_out <- "C:/Users/nabee/New folder/IAP/arima_out_fors2"
dir.create(arima_root_out, recursive = TRUE, showWarnings = FALSE)

fs <- 985
SUBJECTS_MAX <- 2
N_CASES_MAX <- 6
MIN_LEN_STAT <- 200
MIN_LEN_FIT <- 100
MIN_VAR <- 1e-12

clean_vec <- function(x) { x <- as.numeric(x); x <- x[is.finite(x)]; x - mean(x,
na.rm=TRUE) }

make_cases <- function(x, n_cases = N_CASES_MAX, min_len = MIN_LEN_STAT) {

```

```

n <- length(x); out <- list()
for (k in 0:(n_cases-1)) {
  m <- floor(n / (2^k))
  if (m < min_len) break
  out[[paste0("case_", k+1, "_N", m)]] <- x[seq_len(m)]
}
out

adf_p <- function(x) tryCatch(adf.test(x)$p.value, error=function(e) NA_real_)
kpss_p <- function(x) tryCatch(kpss.test(x, null="Level")$p.value, error=function(e) NA_real_)

stationarity_flag <- function(p_adf, p_kpss) {
  if (is.na(p_adf) || is.na(p_kpss)) return("unknown")
  if (p_adf < 0.05 && p_kpss > 0.05) return("stationary")
  if (p_adf < 0.10 && p_kpss > 0.05) return("near-stationary")
  "non-stationary"
}

diff_order_needed <- function(x) {
  d1 <- tryCatch(ndiffs(x, test="adf"), error=function(e) 0L)
  d2 <- tryCatch(ndiffs(x, test="kpss"), error=function(e) 0L)
  max(as.integer(d1), as.integer(d2), na.rm = TRUE)
}

plot_ts_acf_pacf <- function(x, title="Series", fs=fs) {
  df <- data.frame(t = seq_along(x)/fs, y = x)
  p1 <- ggplot(df, aes(t, y)) + geom_line() +
    labs(title=title, x="Time (s)", y="Amplitude") + theme_minimal()
  acf_obj <- Acf(x, plot=FALSE)
  pacf_obj <- Pacf(x, plot=FALSE)
  df_acf <- data.frame(lag = acf_obj$lag[,1,1], acf = acf_obj$acf[,1,1])
  df_pacf <- data.frame(lag = pacf_obj$lag, pacf = pacf_obj$acf)
  p2 <- ggplot(df_acf, aes(lag, acf)) + geom_bar(stat="identity") +
    labs(title="ACF", x="Lag", y="ACF") + theme_minimal()
  p3 <- ggplot(df_pacf, aes(lag, pacf)) + geom_bar(stat="identity") +
    labs(title="PACF", x="Lag", y="PACF") + theme_minimal()
  grid.arrange(p1, p2, p3, ncol=1)
}

plot_fit_overlay <- function(x, fitvals, title="Original vs ARIMA fitted", fs=fs) {
  df <- data.frame(t = seq_along(x)/fs, orig = as.numeric(x), fit = as.numeric(fitvals))
  ggplot(df, aes(t)) +
    geom_line(aes(y=orig), linewidth=0.4) +
    geom_line(aes(y=fit), linewidth=0.4, linetype="dashed") +
    labs(title=title, x="Time (s)", y="Amplitude",
         subtitle="Solid: original noise, Dashed: ARIMA fitted (in-sample)") +
    theme_minimal()
}

plot_resid_diag <- function(fit, title="Residual diagnostics", fs=fs) {
  r <- residuals(fit)
  df <- data.frame(t = seq_along(r)/fs, r = as.numeric(r))
  p1 <- ggplot(df, aes(t, r)) + geom_line() +
    labs(title=paste0(title, ": residuals"), x="Time (s)", y="resid") +
    theme_minimal()
  acf_obj <- Acf(r, plot=FALSE)
  df_acf <- data.frame(lag = acf_obj$lag[,1,1], acf = acf_obj$acf[,1,1])
  p2 <- ggplot(df_acf, aes(lag, acf)) + geom_bar(stat="identity") +
    labs(title="Residual ACF", x="Lag", y="ACF") + theme_minimal()
  lb <- tryCatch(Box.test(r, lag = min(20, floor(length(r)/4)), type="Ljung-Box"),
                 error=function(e) NULL)
  lb_txt <- if (is.null(lb)) "LB: n/a" else sprintf("Ljung-Box p = %.4f", lb$p.value)
  grid.arrange(p1, p2, top = lb_txt)
}

arima_equation_string <- function(fit) {
  ord <- arimaorder(fit); p <- ord[1]; d <- ord[2]; q <- ord[3]
  coefs <- coef(fit)
  phi <- coefs[grep("Aar", names(coefs))]
  theta <- coefs[grep("Ama", names(coefs))]
  c0 <- coefs[grep("mean|intercept|drift", names(coefs))]
  left <- if (length(phi)) paste0("(1 - ", paste(sprintf("%.4f L^%d", phi,
  seq_along(phi))), collapse=" - "), ")") else "(1"
  diffpart <- if (d > 0) paste0("(1 - L)^", d) else "(1"

```

```

    right <- if (length(theta)) paste0("(1 + ", paste(sprintf("%.4f L^%d", theta,
seq_along(theta)), collapse=" + "), ")e_t") else "e_t"
    const <- if (length(c0)) sprintf(" + %.4f", c0[1]) else ""
    paste0(left, " ", diffpart, " x_t = ", right, const)
}

fit_arima_pipeline <- function(x, series_tag, out_dir) {
  x <- clean_vec(x)
  if (length(x) < MIN_LEN_STAT) {
    return(list(status="skipped_too_short_pre", note=sprintf("N=%d < %d", length(x),
MIN_LEN_STAT)))
  }
  if (var(x, na.rm=TRUE) < MIN_VAR) {
    return(list(status="skipped_constant_pre", note="near-constant series before
differencing"))
  }
  p_adf0 <- adf_p(x); p_kpss0 <- kpss_p(x); stat0 <- stationarity_flag(p_adf0,
p_kpss0)
  d_need <- if (stat0 == "stationary") 0L else diff_order_needed(x)
  d_need <- max(0L, min(2L, as.integer(d_need)))
  fit <- tryCatch(
    auto.arima(x, d = d_need, stepwise = FALSE, approximation = FALSE,
               seasonal = FALSE, allowmean = TRUE, allowdrift = TRUE, biasadj =
FALSE),
    error=function(e) NULL
  )
  if (is.null(fit)) {
    return(list(status="fit_error", note="auto.arima failed",
               padf=p_adf0, pkpss=p_kpss0, stat0=stat0, d_need=d_need))
  }
  x_st <- if (d_need > 0) diff(x, differences = d_need) else x
  p_adf1 <- adf_p(x_st); p_kpss1 <- kpss_p(x_st); stat1 <- stationarity_flag(p_adf1,
p_kpss1)
  fitvals <- fitted(fit)
  resid <- residuals(fit)
  rmse_resid <- sqrt(mean(resid^2, na.rm=TRUE))
  aicc <- fit$aicc
  ord <- arimaorder(fit)
  png(file.path(out_dir, paste0(series_tag, "_ts_acf_pacf.png")), width=1200,
height=1600)
  plot_ts_acf_pacf(x, title=paste0(series_tag, " (orig)"), fs=fs); dev.off()
  png(file.path(out_dir, paste0(series_tag, "_overlay_fit.png")), width=1400,
height=900)
  print(plot_fit_overlay(x, fitvals, title=paste0(series_tag, " - original vs ARIMA
fitted"), fs=fs)); dev.off()
  png(file.path(out_dir, paste0(series_tag, "_resid_diag.png")), width=1200,
height=900)
  plot_resid_diag(fit, title=series_tag, fs=fs); dev.off()
  eq_str <- arima_equation_string(fit)
  writeLines(c(
    paste0("Series: ", series_tag),
    paste0("Order: (p,d,q) = (", paste(ord, collapse=","), ")"),
    paste0("AICc: ", sprintf("%.4f", aicc)),
    paste0("RMSE(residuals): ", sprintf("%.6g", rmse_resid)),
    paste0("Stationarity before: ADF p=", sprintf("%.3g", p_adf0),
           ", KPSS p=", sprintf("%.3g", p_kpss0), " -> ", stat0),
    paste0("Differencing used: d=", d_need),
    paste0("Stationarity after differencing: ADF p=", sprintf("%.3g", p_adf1),
           ", KPSS p=", sprintf("%.3g", p_kpss1), " -> ", stat1),
    "", "ARIMA equation (backshift L):", eq_str
  ), con = file.path(out_dir, paste0(series_tag, "_equation.txt")))
  list(
    status="ok",
    padf=p_adf0, pkpss=p_kpss0, stat0=stat0,
    d_need=d_need,
    padf_after=p_adf1, pkpss_after=p_kpss1, stat1=stat1,
    order=ord, aicc=aicc, rmse_resid=rmse_resid
  )
}

all_noise <- list.files(noise_root, pattern="^multich_*_NOISE\\.csv$", full.names
= TRUE, recursive = TRUE)
if (length(all_noise) == 0) stop("No NOISE csv files found under: ", noise_root)

noise_dirs <- dirname(all_noise)
subject_level <- sub(paste0("^", gsub("\\\\\\\\", "/", noise_root), "/?"), "", gsub("\\\\\\\\", "/", noise_dirs))

```

```

subject_level <- sub("./.*$", "", subject_level)
subjects_ord <- unique(subject_level)
subjects_take <- subjects_ord[seq_len(min(SUBJECTS_MAX, length(subjects_ord)))]
keep_mask <- subject_level %in% subjects_take
multi_files <- all_noise[keep_mask]

cat("Subjects selected:", paste(subjects_take, collapse=", "), "\n")
cat("Files to process:", length(multi_files), "\n")

for (nf in multi_files) {
  cat("\n==== Processing:", nf, "====\n")
  dat <- as.matrix(fread(nf)); storage.mode(dat) <- "double"
  if (nrow(dat) < ncol(dat)) dat <- t(dat)
  c <- ncol(dat)
  rel_dir <- dirname(nf)
  rel_tail <- sub(paste0("^", gsub("\\\\", "/", noise_root)), "", gsub("\\\\", "/", rel_dir))
  out_dir_file <- file.path(arima_root_out, rel_tail,
  tools::file_path_sans_ext(basename(nf)))
  dir.create(out_dir_file, recursive = TRUE, showWarnings = FALSE)
  summary_rows <- list()
  for (ch in seq_len(c)) {
    x_full <- clean_vec(dat[, ch])
    if (var(x_full, na.rm=TRUE) < MIN_VAR) {
      cat(sprintf(" Skipping ch%d (near-constant)\n", ch)); next
    }
    cases <- make_cases(x_full, n_cases = N_CASES_MAX, min_len = MIN_LEN_STAT)
    for (nm in names(cases)) {
      x <- cases[[nm]]
      case_tag <- paste0("ch", ch, "_", nm)
      case_dir <- file.path(out_dir_file, paste0("ch", ch), nm)
      dir.create(case_dir, recursive = TRUE, showWarnings = FALSE)
      res <- fit_arima_pipeline(x, series_tag = case_tag, out_dir = case_dir)
      if (!identical(res$status, "ok")) {
        cat(sprintf(" [%-20s] SKIPPED: %s (%s)\n",
                   case_tag, res$status, if (!is.null(res$note)) res$note else ""))
        summary_rows[[length(summary_rows)+1]] <- data.frame(
          file = basename(nf), channel = paste0("ch", ch), case = nm, N = length(x),
          status = res$status, note = if (!is.null(res$note)) res$note else
NA_character_,
          stringsAsFactors = FALSE
        )
        next
      }
      cat(sprintf(" [%-20s] OK ARIMA(%d,%d,%d) RMSE=% .6g AICc=% .2f (before: %s,
after: %s)\n",
                  case_tag, res$order[1], res$order[2], res$order[3],
                  res$rmse_resid, res$aicc, res$stat0, res$stat1))
      summary_rows[[length(summary_rows)+1]] <- data.frame(
        file = basename(nf), channel = paste0("ch", ch), case = nm, N = length(x),
        status = res$status,
        adf_p_before = res$padf, kpss_p_before = res$pkpss, stationarity_before =
res$stat0,
        d_needed = res$d_need,
        adf_p_after = res$padf_after, kpss_p_after = res$pkpss_after,
        stationarity_after = res$stat1,
        arima_p = res$order[1], arima_d = res$order[2], arima_q = res$order[3],
        aicc = res$aicc, rmse_resid = res$rmse_resid,
        stringsAsFactors = FALSE
      )
    }
  }
  if (length(summary_rows)) {
    sum_dt <- data.table::rbindlist(summary_rows, use.names = TRUE, fill = TRUE)
    fwrite(sum_dt, file.path(out_dir_file, "arima_summary.csv"))
    cat(" -> Saved summary:", file.path(out_dir_file, "arima_summary.csv"), "\n")
  }
}

```