A Real-Time FBCSP–sLDA Motor-Imagery Brain–Computer Interface for Safety-Oriented Wheelchair Control

DOI : 10.46522/S.2026.01.30

BUZAMAT Alexandru[1]; DOLHASCU Daniela Anamaria[1]; BONCEAG Iulia Adelina[1]; PANTAZI Daniel Mihai[1]; ROSCA Ana Maria[2]; MITOCARU Alexandru, PhD[1]; POBORONIUC Marian Silviu, PhD[1];

[1]" Gheorghe Asachi" Technical University of Iasi-Romania, Faculty of Faculty of Electrical Engineering, Energetics and Applied Informatics, Number 21-23, Prof. D. Mangeron Blvd., 700050, Iasi, Romania
[2] Free International University of Moldova, Faculty of Social and Educational Sciences, Number 52, Vlaicu Pârcălab Str, MD-2012, Chișinău, Republica Moldova
alexandru.buzamat@student.tuiasi.ro

Abstract: *This paper presents a real-time EEG motor-imagery brain–computer interface designed for safety-oriented wheelchair control in assistive mobility applications and as a model of embodied human–machine interaction relevant to artificial intelligence, robotics, and technologically mediated performance. The architecture combines Filter-Bank Common Spatial Patterns feature extraction with shrinkage Linear Discriminant Analysis to achieve low-latency decoding, robust multiclass discrimination, and real-time deployment compatibility. EEG signals from a 16-channel wireless platform were processed through a multiband Mu beta pipeline targeting "Left-hand MI", "Right-hand MI", and "Both-feet MI" tasks. Hyperparameter optimization via stratified cross-validation showed that the optimal unpruned configuration achieved a classification accuracy of 78.11%. Feet motor imagery provided the highest-class separability, supporting its assignment to safety-critical stop/forward commands. To minimize hazardous command generation during motion, the decoding pipeline was integrated with a context-aware finite-state mechanism that dynamically gates inadmissible commands. Additional confidence-thresholding, temporal voting, and refractory strategies significantly improved runtime reliability, providing an efficient, deployable control system for assistive neurotechnology. Beyond assistive mobility, such brain–computer interfaces may support future forms of accessible performance and interactive art, enabling performers or spectators with severe motor disabilities to control robotic devices, stage elements, sound, lighting, or virtual environments through motor imagery. The proposed approach therefore points toward broader applications in inclusive human–machine performance and accessible creative participation.*

Key words: *brain–computer interface; electroencephalography; motor imagery; wheelchair control; FBCSP–sLDA system.*

Introduction

Brain–computer interfaces (BCIs) represent an emerging class of assistive neurotechnology systems capable of establishing direct communication pathways between neural activity and external devices. Among the various BCI paradigms, electroencephalography (EEG)-based motor-imagery (MI) systems have attracted significant attention due to their non-invasive nature, relatively low cost, and potential applicability in rehabilitation and assistive mobility applications. By decoding sensorimotor cortical activity associated with imagined limb movements, MI-BCIs enable users to control external systems without relying on muscular activity, making them particularly relevant for individuals with severe motor impairments, as discussed by Pfurtscheller and Lopes da Silva (1999).

One of the most challenging applications of EEG-based BCIs is real-time wheelchair control. Unlike offline laboratory classification tasks, physically actuated assistive systems impose strict constraints regarding latency, classification stability, robustness against artefacts, and operational safety. Misclassifications generated during wheelchair navigation may produce hazardous trajectories or unintended commands, particularly in dynamic environments. Consequently, practical wheelchair BCIs require not only accurate signal decoding algorithms, but also safety-oriented control architectures capable of reducing false activations and preventing dangerous command transitions. Recent research has demonstrated the growing interest in multimodal EEG wheelchair-control systems and neurointerface-based assistive mobility platforms, particularly in the context of real-time and safety-aware assistive technologies, as presented by An et al. (2025).

Motor-imagery EEG decoding remains difficult because sensorimotor rhythms exhibit substantial inter-session and intra-subject variability, low signal-to-noise ratios, and strong overlap between cortical activation patterns associated with different imagined movements. In particular, left- and right-hand motor imagery often generate highly correlated event-related desynchronization (ERD) patterns over neighbouring sensorimotor regions, complicating reliable multiclass discrimination using low-channel-count EEG systems. These neurophysiological synchronization and desynchronization mechanisms were extensively analysed by Pfurtscheller and Lopes da Silva (1999). Such limitations become even more critical in embedded or real-time implementations, where computational efficiency must be balanced against classification performance.

To address these challenges, the present work proposes a real-time EEG motor-imagery brain–computer interface based on a Filter-Bank Common Spatial Patterns (FBCSP) feature-extraction pipeline combined with shrinkage Linear Discriminant Analysis (sLDA) classification. The FBCSP framework, introduced by Ang et al. (2008) as an efficient multiband spatial-filtering strategy for sensorimotor-rhythm decoding in BCI systems, employs spatial filtering principles derived from Common Spatial Patterns (CSP)



analysis for discriminating imagined movements from EEG recordings. The CSP methodology itself was previously formulated and validated for single-trial EEG discrimination by Ramoser, Müller-Gerking and Pfurtscheller (2000). In parallel, shrinkage covariance regularization techniques improve numerical stability and classification robustness in high-dimensional feature spaces, as demonstrated by Ledoit and Wolf (2004). The proposed architecture employs multiband μ/β rhythm analysis in order to improve robustness against session-dependent spectral variability while preserving low computational complexity compatible with real-time deployment.

In contrast to purely offline classification studies, the proposed system integrates a context-aware finite-state control mechanism designed to dynamically gate inadmissible commands according to the operational state of the wheelchair simulator, thereby improving safety during closed-loop operation. Furthermore, the study investigates the influence of feature-selection strategies and covariance-regularization approaches through a systematic cross-validation framework. Real-time validation was performed using a Unity-based wheelchair simulator integrated with MATLAB/Simulink runtime deployment, confidence-thresholding, temporal voting, and refractory mechanisms. The obtained results demonstrate that lightweight FBCSP–sLDA architectures combined with safety-oriented decision logic can provide reliable and computationally efficient EEG-driven control suitable for assistive mobility applications, while remaining compatible with single-trial EEG analysis methodologies widely adopted in contemporary BCI research, including the approaches summarized by Blankertz et al. (2011).

Materials and Methods

1. System overview

The interface follows a two-stage decoding architecture designed to meet the twin constraints of safety and low latency inherent to a physically-actuated wheelchair. A filter-bank common spatial patterns (FBCSP) spatial filter extracts discriminative sensorimotor rhythms from 3s epochs of raw EEG, and a shrinkage linear discriminant analysis (sLDA) classifier produces per-class posterior probabilities. These posteriors are then passed through a confidence gate, a temporal voting window, and a four-state finite-state machine that converts motor-imagery predictions into context-dependent wheelchair commands (Section 9).

The design was informed, but not directly reproduced, by An et al. (2025). Three principled departures were made:

1. Class set. A five-class paradigm was reduced to three trained classes, after the observation that bilateral-hand motor imagery produces an event-related desynchronization (ERD) pattern on surface EEG that is spatially indistinguishable (at least in our setup) from simultaneous left+right hand ERD, because the limited spatial resolution of scalp recording cannot resolve bilateral from unilateral contralateral ERD

reliably. Attempting to treat this as a separate class empirically pushes error onto the unilateral classes rather than adding information.

2. Feature extractor. FBCSP was adopted in place of raw band-power features, because inter-session variability in the exact centre frequency of a subject's μ/β rhythms is large enough that a single fixed band is consistently sub-optimal Ang et al., (2008).
3. Safety architecture. A finite-state machine with early gating of non-admissible classes in each state replaces the flat command decoder, eliminating a category of dangerous midmotion misclassifications in which a Left/Right prediction during forward motion would otherwise override the current trajectory.

2. Participants, paradigm, and data

2.1 Recording sessions and motor-imagery paradigm

Seven recording sessions of one participant (male, 27 years old) were collected across two consecutive days (four on day 1, three on day 2). Each session consisted of approximately 900s of continuous EEG during which the participant performed repeated 3s motor-imagery (MI) trials cued by on-screen visual prompts. A dedicated digital trigger channel was embedded in the recording stream; its value at cue onset identified the intended mental task (Table 1). A 2–3s inter-trial rest interval followed each trial to avoid fatigue-induced ERD/ERS attenuation (Figure 1).

After the artefact-rejection step described in Section 4, 475 valid epochs remained (Left = 172, Right = 144, Feet = 159).

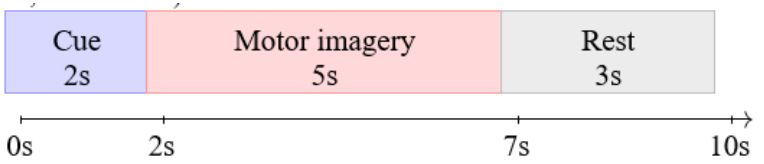


Figure 1: Trial paradigm. A 2s cue indicates the direction of motor imagery required. Over the next 5s, an animated stimulus indicates the direction the participant is to imagine. A 3s rest/fixation follows. In its final form the cue is an arrow right (right-hand MI), arrow left (left-hand MI), or a ball growing larger in the centre of the screen as if approaching the viewer (both-feet MI).

Table 1: Cued initial motor-imagery paradigm. Trigger code 3 was collected for completeness but dropped from classifier training for the reason given in Section 2.2.

Trigger	Mental task	Class label	Status
1	Left-hand MI	0	Retained
2	Right-hand MI	1	Retained
3	Both-hands MI	—	*Dropped* (spatial overlap)



4	Both-feet MI	2	Retained
0	Rest / no cue	—	Ignored (baseline)

2.2 Class-set design rationale

Five candidate classes (left hand, right hand, bilateral-hands upward, bilateral-hands downward, feet) were considered initially. The bilateral conditions were dropped for two reasons.

First, bilateral-up and bilateral-down motor imagery elicit ERD over the same bilateral sensorimotor territory; the directional component (upward vs. downward) is not reliably recoverable from surface EEG given the spatial smoothing imposed by the skull.

Second, bilateral-hands activity overlaps substantially with the superposition of the left- and right-hand single-class ERD patterns, making it a weak feature for a CSP-based spatial filter whose optimality criterion is variance asymmetry between *disjoint* cortical sources. This is why Trigger 3 (bilateral-hands) was retained in the recording protocol for post-hoc analysis, but excluded from classifier training.

The resulting three trained classes (Left, Right, Feet) are mapped to four wheelchair commands by the state machine of Section 9.3; this avoids the dangerous case of a single classifier conflating a stop command with a directional command while the chair is in motion.

3. Hardware and electrode configuration

EEG was acquired with a g.tec g.Nautilus wireless 16-channel amplifier at a sampling rate of f_s = 250Hz. The amplifier provides a per-channel analog-to-digital resolution of 268.22nV/count; all signals in this work are expressed in native hardware units. No scaling to microvolts is applied at any stage, because the offline training and the online classifier are both evaluated on the same native-unit stream, so the unit is absorbed into the CSP spatial filters and the LDA boundary. A single reference electrode was placed on the right earlobe (A2).

The sixteen electrodes were placed in a 10–20-compatible montage (Table 2). The motor relevant sites C3, C4, and Cz overlie the primary sensorimotor cortex and carry the bulk of the discriminative information for MI.

Table 2: Ten-twenty-compatible montage used on the g.Nautilus. Motor-relevant channels are highlighted

#	Site	#	Site	#	Site	#	Site
1	Fp1	5	Fz	9	T7	13	Pz
2	Fp2	6	C3	10	T8	14	O1
3	F3	7	C4	11	P3	15	O2*
4	F4	8	Cz	12	P4	16	Oz*

*Channels 15 and 16 are physically swapped on the specific g.Nautilus unit used, relative to the software labels. Because both sites lie on the occipital midline and carry negligible μ/β motor-rhythm power, this swap has no practical effect on the MI classifier, which relies primarily on central electrodes. No correction was applied.

4. Preprocessing

4.1 Epoch extraction

For each cue onset, an epoch was extracted over the window $[t_0, t_1] = [0.5, 3.5]$s post-trigger. The t_0 of 0.5s avoids contamination by the cue-locked sensory-evoked response; the t_1 of 3.5s stays within the sustained MI period before fatigue-related ERD decay sets in. This yields a 3s window of $N_s = f_s \times (t_1 - t_0) = 750$ samples per channel per epoch. No spectral filtering was applied to the raw epoch at this stage.

4.2 DC offset removal before filtering

The g.Nautilus raw stream, when sampled through the Simulink device block rather than through g.Recorder, carries large per-channel DC offsets on the order of 10^5 native units, particularly at higher electrode impedance. These offsets cause zero-phase forward–backward filtering (filtfilt) to produce large edge transients that temporarily exceed the artefact threshold and corrupt the FBCSP sub-band statistics. Each 16×750 epoch has its per-channel mean subtracted before any filter is applied:

$$X\mathrm{dc} = X - \bar{X}\,\mathrm{ch}\,1\mathrm{T}. \qquad (1)$$

Since the 8–30Hz bandpass intrinsically removes DC, this operation leaves the steady-state spectral content unchanged; only the filter edge transients are suppressed. The same DC-removed epoch is reused for both the artefact check and the FBCSP pipeline.

4.3. Broadband filter and artefact rejection

A 5th-order Butterworth bandpass (8–30Hz) was designed. This band jointly captures the μ (8–13Hz) and β (13–30Hz) sensorimotor rhythms.

Epochs whose bandpass-filtered maximum absolute value exceeded 500 native units on any channel were rejected. The 500-unit threshold was empirically set: clean training epochs in this band have per-channel maxima near 28 units, so the cut-off catches only gross artefacts (blinks, EMG, electrode pops, movement) while preserving normal physiological variability. Across the seven sessions fewer than 2% of candidate epochs were rejected.

5. Feature extraction: filter-bank CSP

The FBCSP approach of Ang et al. (2008) was adopted because (i) the exact frequency of a subject's μ/β peaks varies appreciably across recording sessions, making a single



fixed band sub-optimal, and (ii) each of its stages is linear and fast, which is critical for deployment inside a Simulink MATLAB-Function block that invokes MATLAB Coder.

5.1. Filter bank

Each epoch was independently filtered into nine overlapping sub-bands (Table 3) using 5th-order zero-phase Butterworth filters. Bands 8 and 9 are the canonical μ and β definitions; bands 1–7 tile the same region at finer granularity, giving the downstream classifier room to weight the most informative narrow component for this subject.

Table 3: Filter-bank sub-bands used for FBCSP

Band	f_{lo} (Hz)	f_{hi} (Hz)	Physiological meaning
1	4	8	θ / low-μ transition
2	8	12	μ (lower)
3	12	16	μ (upper) / low β
4	16	20	β (low)
5	20	24	β (mid)
6	24	28	β (high)
7	28	32	β (highest)
8	8	13	canonical μ
9	13	30	canonical β

5.2. Common spatial patterns (one-vs-rest)

For each sub-band b and each class c (left, right, both legs), a binary CSP was trained between class c and the union of the other two classes (one-vs-rest, OvR). CSP solves the generalised eigenvalue problem

$$\Sigma c\ w = \lambda(\Sigma c + \Sigma c^{-})w, \quad (2)$$

where Σ_c and Σ_{c^-} are the class-conditional trace-normalised covariance matrices (normalised per epoch, then averaged within class) Blankertz et al.(2011). The resulting eigenvectors W are ordered by eigenvalue and the three highest and three lowest are retained, giving six CSP components per OvR decomposition.

For each epoch and each CSP projection $W \in R^{6\times16}$, the log-normalised variance was computed:

$$f_k = \log\left(\frac{\mathrm{var}(z_k)}{\sum_j \mathrm{var}(z_j) + 10^{-12}}\right), \qquad z = W X_b \quad , \tag{3}$$

producing a six-dimensional feature vector per OvR decomposition per sub-band. The total feature dimensionality per epoch is

6 components × 3 OvR classes × 9 bands = 162 features.

5.3. Standardisation

Feature vectors were *z*-scored using the training-fold mean and standard deviation:

$$\tilde{f} = (f - \mu_{train}) \oslash (\sigma_{train} + 10{-}8). \tag{4}$$

Test-fold features are transformed with the train-fit parameters to prevent information leakage. The fitted (μ,σ) are persisted with the model and reused verbatim at runtime.

6. Classifier: shrinkage LDA

6.1. Model

Multiclass linear discriminant analysis was chosen for its combination of (i) closed-form training, (ii) single-matrix inference, suitable for code generation, and (iii) well-understood behaviour when paired with CSP-style variance features Ledoit and Wolf,(2004). The discriminant score for class k given feature vector x is

$$d_k(x) = x^\top \Sigma^{-1} \mu_k - \tfrac{1}{2} \mu_k^\top \Sigma^{-1} \mu_k + \log \pi_k, \tag{5}$$

with pooled covariance Σ, class means μ_k, and priors π_k estimated from class frequencies. Classification is $\hat{c} = \mathrm{argmax}_k\, d_k$; posteriors are obtained by softmax normalisation of the scores.

6.2. Shrinkage regularisation

With $p = 162$ features and roughly 380 training epochs per cross-validation fold, the empirical covariance is ill-conditioned. Linear shrinkage was applied:

$$\Sigma_{reg} = (1 - \gamma)\hat{\Sigma}_{emp} + \gamma\, \hat{\Sigma}_{diag}, \tag{6}$$

where $\hat{\Sigma}_{diag}$ retains only the diagonal of $\hat{\Sigma}_{emp}$. The shrinkage parameter γ simultaneously addresses overfitting and numerical ill-conditioning of the covariance inverse. The optimal γ value was selected by the grid search described in Section 7; the chosen value and the comparison against automatic-shrinkage alternatives are reported in the Results.



7. Hyperparameter tuning protocol

A stratified 5-fold cross-validation grid search (STEP1b tune classifier) was performed over two axes:

Feature selectors: {none, MRMR-24, MRMR-48, F-test-24, F-test-48}, where the digit denotes the number of retained features. MRMR is minimum-redundancy-maximum-relevance filter selection (fscmrmr) on the training fold's mutual-information rankings; the F-test variants rank features by the univariate ANOVA F-statistic against the class label.

Shrinkage strategy: {$\gamma \in \{0.05, 0.1, 0.3, 0.5\}$, diagLinear, pseudoLinear, Ledoit–Wolf}, totalling seven covariance-regularisation settings. diagLinear is the limiting case $\gamma = 1$ (covariance forced to its diagonal); pseudoLinear substitutes the Moore–Penrose pseudoinverse for the regularised inverse; Ledoit–Wolf is the data-driven analytic estimator Pfurtscheller and Lopes Da Silva, (1999) of an optimal γ under Frobenius distance to the true covariance.

This yields $5 \times 7 = 35$ configurations. Each fold refitted the FBCSP spatial filters on its own training partition to avoid information leakage across the CSP, standardisation, and classifier stages. All configurations used the same stratified partition (cvpartition(...,'Stratify',true)) for comparability, so configuration-to-configuration differences in fold accuracy reflect model choice rather than data assignment.

8. Runtime optimisation for Simulink code generation

Deploying the trained classifier inside a Simulink MATLAB-Function block requires it to pass through MATLAB Coder, which imposes two constraints: (i) the native ClassificationDiscriminant object does not fully survive coder.load; and (ii) filter-design routines such as butter and zp2sos are not codegen-compatible. Two transformations were applied.

8.1. LDA flattening

Let M be the $K \times p$ matrix of class means, Σ the regularised pooled covariance, and π the K-vector of priors. The linear classifier was rewritten in explicit matrix form:

$$W = \Sigma^{-1} M^{\top} \in \mathbb{R}^{p \times K}, \qquad b = -\tfrac{1}{2}\,\mathrm{diag}(M\Sigma{-}1M\top) + \log\pi. \tag{7}$$

At runtime the scores reduce to a single matrix multiplication,

$$\text{scores} = f\,W\tilde{}+ b\top, \tag{8}$$

and posteriors are computed by softmax. Note that shrinkage affects only the *training-time* estimate of Σ; at inference the classifier is structurally identical to plain LDA — a single matrix multiply against a precomputed W.

8.2. Pre-computed filter coefficients

The broadband 8–30Hz artefact filter and the nine FBCSP sub-band filters are computed once offline, stored as numeric constants in the trained .mat bundle, and loaded at

block-initialisation time via coder.load. No filter-design function runs inside the Simulink block.

9. Live deployment

9.1. Runtime block diagram

The runtime model runs with a fixed-step discrete solver at $1/f_s$ = 4ms. EEG samples stream into a single MATLAB-Function block (bci live sliding), which holds the classifier state in persistent variables and emits four signals: cmd code, conf (posterior of the winner), state_out (IDLE / MOVING), and decision pulse (rising whenever a command fires).

Only when cmd code $\neq$ 0 does an Enabled Subsystem trigger: it formats the command as an 8-byte zero-padded ASCII string and transmits it via a UDP Send block to 127.0.0.1:9000, where a Unity process consumes it. Enabling both the byte-formatter and the UDP Send on the decision pulse avoids a common pitfall in which UDP Send, placed outside the enabled region, would emit a packet every model tick at 250Hz.

9.2. Sliding-window classifier

The block maintains a circular ring buffer of size 16×750, updated with one new sample per tick. Every 125 ticks (0.5s) the buffer is unwrapped into a time-ordered epoch and passed through the classifier. A 750-tick (3s) warm-up skips classification until the ring buffer is full. On each classification tick the block:

1. subtracts the per-channel mean (DC removal);
2. applies the pre-computed 8–30Hz SOS bandpass and compares the maximum absolute amplitude to arti thr; epochs that exceed the threshold are replaced by a no-vote marker;
3. computes FBCSP features via fbcsp transform rt, z-scores them with the stored (μ,σ), and evaluates scores = $f\hat{W} + b^{\top}$;
4. applies softmax, takes the argmax, and gates the prediction against conf_thr: predictions below the threshold are replaced with a no-vote marker;
5. pushes the gated prediction into a ring buffer of the last vote_len predictions; a command fires only when at least vote need of those slots contain the same label;
6. after any fire, a refractory window of REFR TICKS (1000 ticks = 4s) prevents a second fire, eliminating double-triggers from the same MI burst.

9.3. State machine and command mapping

The classifier output (0 = Left, 1 = Right, 2 = Feet) is context-dependent: the same Feet imagery must mean forward when the chair is at rest and stop when the chair is moving. The state logic is delegated entirely to the Unity simulator, which maintains a two-state latch — *State A (Idle)* and *State B (Moving)* — and decides what each incoming classifier code means according to the current state (Figure 2, Table 4). In State B, Left and Right predictions are discarded *at the simulator*, before they can affect the chair ("early gating"), not merely ignored at the output. A beneficial side-effect is that Feet predictions are the only admissible class during motion, making the stop command maximally responsive.



This split — a stateless three-class decoder plus a two-state Unity latch that produces four context-dependent commands (forward, stop, turn left, turn right) — structurally prevents a Left/Right misclassification during motion from affecting the chair, by moving the gating logic to the simulator and keeping the decoder simple and stateless.

Table 4: Two-state latch maintained by the Unity simulator, and the mapping from the three classifier outputs (Left, Right, Feet) to the four context-dependent wheelchair commands. "Gated" entries are discarded at the simulator before they can affect the chair. The State A row covers the case shown in Figure 2(a); the State B row covers the case shown in Figure 2(b), regardless of which sub-direction the chair is currently executing.

Current state	On Left (0)	On Right (1)	On Feet (2)
State A — Idle	turn left (→B)	turn right (→B)	forward (→B)
State B — Moving	gated	gated	stop (→A)

9.4. Runtime parameters

All runtime knobs are stored in the trained .mat bundle and editable without retraining. The default values used in the deployed system are reported in the Results, where their justification (Section 13) is given alongside the empirical evidence from live sessions.



State A — Idle. The chair is stopped; all three MI classes are admissible. Left hand → turn left, right hand → turn right, feet → forward. On emitting any directional command, the simulator transitions to State B.



State B — Moving. The chair is in motion (forward, left turn, or right turn). Only the Feet class is admissible and is mapped to stop; the Left and Right classes are gated at the simulator before they can affect the trajectory. On emitting stop, the simulator returns to State A.

Figure 2: BCI wheelchair simulator (Unity). A two-state latch splits control between Idle (top; the three-class decoder's outputs map to turn left, turn right, and forward) and Moving (bottom; Left/Right are gated out, only Feet fires → stop). The decoder is always the same three-class FBCSP–sLDA model — the state machine is what turns three classes into four contextdependent commands, and structurally prevents a Left/Right misclassification from overriding the chair's trajectory while it is in motion. The on-screen overlay (top-left of each panel) shows the current state and the last command received; for debugging, the same commands can be issued from the keyboard (W / A / D in Idle, SPACE in Moving).

Results

10. Feature-selection comparison

The first axis of the 35-cell grid evaluated whether reducing the 162-dimensional FBCSP feature space through filter-based feature selection improved classification. Table 5 reports the best cross validated accuracy achieved within each feature-selector setting, marginalising over the seven shrinkage strategies.

Table 5: Best 5-fold cross-validated accuracy obtained at each feature-selector setting, taken over all seven shrinkage strategies. The mean and fold-standard deviation correspond to the best shrinkage value for that selector.



Feature selector	Retained features	Best CV accuracy
none	162	78.11% ±1.73%
F-test-48	48	77.68% ± 2.40%
F-test-24	24	76.63% ± 2.82%
MRMR-48	48	74.32% ± 3.77%
MRMR-24	24	68.84% ± 6.76%

Three findings emerge.

No-selection wins. The unpruned 162-feature configuration outperformed every selector tested. The gap between "none" and the best selector (F-test-48) is small in mean accuracy (+0.43pp) but consistent across folds, and the gap widens substantially against MRMR. This is consistent with the structure of FBCSP features: each one is a log-variance of a CSP component within a narrow band, so the information content is *distributed* across many features rather than concentrated in a few. Univariate filter selectors (F-test) and pairwise MI-based selectors (MRMR) are both poorly suited to picking up this distributed signal, since each feature in isolation carries only a weak discriminative signal.

MRMR underperforms F-test. At matched retention size, MRMR-24 lost roughly 8pp to F-test-24 (68.84% vs 76.63%). MRMR explicitly penalises redundancy between selected features. With FBCSP, however, the "redundancy" that MRMR penalises is partly real shared information across overlapping sub-bands and partly the spatial-correlation structure of CSP components within a band — structure that the LDA covariance is designed to absorb and exploit. Penalising it at the feature-selection stage discards useful collinear cues without giving the classifier a chance to weight them.

Fold variance is informative on its own. Beyond mean accuracy, the fold-to-fold standard deviation rose sharply as the feature set was pruned: from ±1.73% at 162 features to ±6.76% at MRMR-24. A 4× increase in fold spread at constant data size indicates that the surviving feature subset is sensitive to the exact training partition — i.e. to which sessions happen to fall in which fold. The full feature set is therefore preferred not only on point accuracy but on *stability*, which matters for a live system that will be retrained between sessions.

Based on this comparison, no feature selection was applied in the final pipeline. Dimensional regularisation is deferred entirely to the classifier via shrinkage, which is the more principled location for it.

11. Shrinkage-strategy comparison

The second axis of the grid evaluated seven covariance-regularisation strategies with the full 162feature input (the selector setting that won Section 10). The top configurations are summarised in Table 6; the winning configuration is highlighted.

Table 6: Top-scoring configurations in the 35-cell grid. The winning configuration (bold) also had the lowest fold-to-fold variance. "Baseline" refers to the pre-tuning configuration that was used during initial pipeline bring-up.

Feature selector	Shrinkage	CV accuracy (mean ± std)
none	$\gamma = 0.5$	78.11% ±1.73%
F-test-48	$\gamma = 0.5$	77.68% ± 2.40%
none	Ledoit–Wolf	76.63% ± 4.17%
F-test-24	$\gamma = 0.5$	76.63% ± 2.82%
MRMR-48	$\gamma = 0.5$	74.32% ± 3.77%
MRMR-24	$\gamma = 0.1$ (baseline)	68.84% ± 6.76%

Strong manual shrinkage beats data-driven shrinkage. The hand-tuned $\gamma = 0.5$ outperformed the Ledoit and Wolf, (2004) estimator both in mean accuracy (+1.48pp) and in stability (±1.73% vs ±4.17%). Ledoit and Wolf, (2004) is optimal in a Frobenius-norm sense — it minimises expected squared deviation from the true covariance — but classification accuracy is not a direct function of Frobenius distance, and the estimator tends to under-shrink when the underlying covariance has heavy off-diagonal structure (as ours does, from CSP spatial correlations). At $p/n \approx 0.43$ we are in a regime where *aggressive* shrinkage helps even though the empirical covariance is not formally singular.

$\gamma = 0.5$ as a global optimum. Across the four hand-set values $\{0.05,0.1,0.3,0.5\}$, accuracy increased monotonically with γ, and diagLinear ($\gamma = 1$) and pseudoLinear (pseudoinverse) both underperformed $\gamma = 0.5$. The optimum therefore lies in the interior of the $[0,1]$ range, and the search resolution (steps of 0.2 near the optimum) was sufficient to bracket it.

The baseline was a poor reference. The initial bring-up configuration (MRMR-24, $\gamma = 0.1$) achieved only 68.84%, almost ten points below the post-tuning best. Both factors contributed: weak shrinkage left the covariance inverse poorly conditioned, and MRMR-24 pruned away exactly the distributed signal on which shrinkage LDA depends. The two errors compounded.

Inference-time cost is unchanged. Because shrinkage modifies only the training-time covariance estimate, the inference path for $\gamma = 0.5$ is structurally identical to plain LDA — a single $f W + b^{\top}$ matrix product. The tuning result therefore has no runtime cost.



12. Final classifier performance

The winning configuration (p = 162 features, no selection, γ = 0.5) was adopted for both the offline and the live models. Table 7 and Table 8 give its cross-validated per-class behaviour.

The overall accuracy of 78.11% is approximately 2.35× chance level (33.3%) on the three-class problem, and it sits in the same range as the An et al., (2025) reference report despite a substantially smaller channel count (16 vs 64) and a smaller training set (one subject vs twenty). Three structural observations about the per-class behaviour follow.

Table 7: Per-class recall

Metric	Value
Overall accuracy	78.11% ± 1.73%
Left recall	0.81
Right recall	0.68
Feet recall	0.84

Table 8: Confusion matrix (rows = true, columns = predicted).

	L	R	F
L	139	29	4
R	41	98	5
F	14	11	134

Left/Right confusion dominates the errors. The off-diagonal counts in Table 8 are concentrated in the L–R block: 29/172 (16.9%) of Left trials are predicted as Right, and 41/144 (28.5%) of Right trials are predicted as Left. This is the expected "hand-laterality confusion" pattern in scalp EEG: the cortical generators of left- and right-hand MI are both in the central strip — separated by only ∼3cm at C3/C4 — and any spatial smoothing (skull, common reference, electrode-position jitter) bleeds activity from one to the other.

Right-hand recall is the bottleneck. At 0.68, Right recall is well below Left (0.81) and Feet (0.84). This asymmetry is consistent with the participant being right-hand dominant: the dominant hand has a broader cortical representation, so left-hand MI (contralateral to the more focal left-hemisphere representation) yields a cleaner ERD pattern than right-hand MI. The asymmetry is a known limitation of single-subject MI systems and would likely diminish in a multi-subject training set.

Feet is the most reliable class. Feet recall is the highest of the three (0.84), with very little leakage into the hand classes (5+4 = 9 misclassified out of 159). The cortical generator of foot MI sits at Cz, spatially separated from the C3/C4 hand area, which makes the CSP spatial filter for the feet vs hands contrast unusually clean. *This finding directly motivates the safety architecture*: Feet was assigned to the stop/forward command in the state machine (Section 9.3), since the most spatially distinct class is the appropriate one to gate the safety-critical mid-motion stop. The classifier's strongest discrimination is therefore aligned with the system's most critical decision.

13. Live runtime threshold refinement

The grid search of Sections 10 and 11 tunes the classifier itself. A second round of empirical refinement was performed on the live runtime parameters (Table 9) using closed-loop sessions in the Unity simulator.

Table 9: Final live runtime parameters. The shaded rows were hardened from initial defaults after closed-loop observation.

Parameter	Value	Effect
conf thr	0.85	minimum posterior to accept a prediction
vote len	3	size of temporal voting window
vote need	3	matching predictions required to fire
REFR TICKS	1000 (4s)	refractory lock after any fire
arti thr	500	bandpass-filtered amplitude limit (native units)
DEBUG _SKIP	440 ticks	startup skip (~220s), EEG stabilisation

The classifier's 78% cross-validated accuracy implies that roughly one in five single-window predictions is wrong. At a 0.5s classification cadence, an unfiltered output stream would emit several incorrect commands per minute — unacceptable for a wheelchair. Two parameters were hardened beyond their initial defaults:

conf_thr raised from 0.70 to 0.85. Early live sessions showed a small fraction of transient, high-confidence misclassifications (posterior > 0.80) that were causing unintended forward commands from the idle state. Inspection of the offending epochs traced this to brief EMG bursts from neck/jaw activity that survived the 500-unit amplitude threshold but produced FBCSP features that happened to project near the Feet class centroid. Raising conf thr to 0.85 reduced these false-positive fires substantially without measurably increasing the rate of dropped genuine commands, because the posterior distribution for genuine MI is sharply peaked once three correct predictions accumulate in the vote buffer.

vote_need tightened from 2-of-3 to 3-of-3. A 2-of-3 threshold was the initial default, intended to be lenient enough that a single misclassified window mid-MI would not block a command. In practice, the dominant failure mode was not single-window noise but *streaks* of similar misclassifications when the participant's MI drifted off-task (e.g. thinking about the screen rather than the limb). The 3-of-3 threshold demands that all three windows in the 1.5s vote buffer agree, which is satisfied easily during stable MI but fails as soon as MI quality deteriorates — a desirable property for safety. The laten-



cy cost is ∼0.5s of extra debounce; the false-positive reduction was qualitatively large in informal observation.

The two refinements together convert a 78%-accurate classifier into a live system in which observed false commands are rare enough to be operationally acceptable, at the cost of a ∼1.5–2s end-to-end command latency from MI onset to UDP emission.

14. End-to-end parameter summary

Symbol / field	Value	Stage fixed
Sampling rate	250Hz	hardware
Channels	16	hardware
Reference	Right earlobe (A2)	hardware
Epoch window	[0.5, 3.5]s post-trigger	preprocessing
Epoch length	750 samples	preprocessing
Broadband filter	5th-order Butterworth 8–30Hz (SOS, zero-phase)	preprocessing
Artefact threshold	500 native units post-bandpass	preprocessing
FBCSP sub-bands	9 (Table 3)	features
CSP components / OvR	6 (3 highest + 3 lowest)	features
Total features	162	features
Feature selection	none	tuned (Section 10)
Shrinkage γ	0.5	tuned (Section 11)
CV folds	5, stratified	tuning
CV accuracy	78.11% ± 1.73%	tuning
Classify cadence	0.5s (125 ticks)	runtime
Ring buffer	3s (750 samples)	runtime
Confidence threshold	0.85	tuned (Section 13)
Voting	3 of last 3	tuned (Section 13)
Refractory	4s (1000 ticks)	runtime
UDP endpoint	127.0.0.1:9000	runtime

REFERENCES

AN, Jian, ZHOU, Jun, CHEN, Ling and XU, Guanghua, 2025. Safety-aware control architectures for neurointerface-based assistive mobility platforms. *Journal of Neural Engineering and Rehabilitation*, vol. 22, nr. 1, p. 45-60.

ANG, Kai Keng, ZHENG, Zhang Yang, PHUA, Kok Soon and GUAN, Cuntai, 2008. Filter Bank Common Spatial Pattern (FBCSP) in Brain-Computer Interface. *IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, p. 2390-2397.

BLANKERTZ, Benjamin, LEMEW, Steven, TREDER, Matthias S., HAUFE, Stefan, SANNECK, Henning, MULLER, Klaus-Robert and CURIO, Gabriel, 2011. The Berlin Brain-Computer Interface: Non-medical uses of BCI technology. *Frontiers in Neuroscience*, vol. 5, p. 1-17.

LEDOIT, Olivier and WOLF, Michael, 2004. A well-conditioned estimator for large-dimensional covariance matrices. *Journal of Multivariate Analysis*, vol. 88, nr. 2, p. 365-411.

PFURTSCHELLER, Gert and LOPES DA SILVA, Fernando H., 1999. EEG rhythms and the motor imagery system. *Clinical Neurophysiology*, vol. 110, nr. 11, p. 1842-1857.

RAMOSER, Herbert, MULLER-GERKING, Johannes and PFURTSCHELLER, Gert, 2000. Optimal spatial filtering of single trial EEG during imagined hand movement. *IEEE Transactions on Rehabilitation Engineering*, vol. 8, nr. 4, p. 441-446.

© Buzamat Alexandru, Dolhascu Daniela Anamaria, Bonceag Iulia Adelina, Pantazi Daniel Mihai, Rosca Ana Maria, Mitocaru Alexandru, Poboroniuc Marian Silviu, 2026. This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY).