

Block-LMS Algorithmus - Testergebnisse

Projekt: DSP Gruppe 4 - Adaptive 50Hz Störunterdrückung

Datum: 17. Januar 2026

Zielplattform: FM4-176-S6E2CC-ETH DSP-Board

Testsignal: test.wav (Sprachsignal mit 50Hz Netzbrummen, 8kHz Abtastrate)

1. Übersicht

Der Block-LMS (Least Mean Squares) Algorithmus wurde zur adaptiven Unterdrückung von 50Hz Netzbrummen in Sprachsignalen implementiert und getestet. Der Algorithmus verwendet ein sinusförmiges Referenzsignal bei der Störfrequenz und passt die Filterkoeffizienten adaptiv an.

Getestete Parameter

Parameter	Getestete Werte
M (Filterordnung)	8, 10, 12, 14, 16, 20, 24, 32
μ (Schrittweite)	0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1
Blockgröße	128 Samples
Störfrequenz	50 Hz
Abtastfrequenz	8000 Hz

2. Ergebnis-Matrix: Störunterdrückung [dB]

M \ μ	0.0001	0.0005	0.001	0.005	0.01	0.05	0.1
8	0.46	1.95	2.50	2.36	0.86	-0.97	0.37
10	0.60	2.20	2.69	2.30	0.93	1.56	3.49
12	0.74	2.41	2.85	2.53	1.77	4.08	6.84
14	0.88	2.60	3.01	3.16	3.21	6.78	-32.02
16	1.02	2.77	3.22	4.16	4.89	9.83	-44.56

20	1.29	3.15	3.92	6.89	8.22	16.21	-49.75
24	1.54	3.72	5.11	9.84	12.02	**21.49**	-52.45
32	2.03	5.68	8.60	16.27	**20.12**	-42.33	-53.10

Hinweis: Negative Werte bedeuten Instabilität (Verstärkung statt Unterdrückung)

3. Beste Parameterkombinationen

Empfehlung 1: M=24, $\mu=0.05$

Störunterdrückung: 21.49 dB

Konvergenzzeit: ~50-100 ms

Stabilität: Gut

Empfehlung 2: M=32, $\mu=0.01$

Störunterdrückung: 20.12 dB

Konvergenzzeit: ~100-200 ms

Stabilität: Sehr gut (konservativer)

Empfehlung 3: M=24, $\mu=0.01$ (für maximale Stabilität)

Störunterdrückung: 12.02 dB

Konvergenzzeit: ~200-300 ms

Stabilität: Exzellent

4. DSP-Board Rechenaufwand-Analyse

Operationen pro Sample

M	sin()/Sample	Mult/Sample	Add/Sample	Gesamt OPs/Sample
8	16	37	35	~88
12	24	53	51	~128
16	32	69	67	~168
20	40	85	83	~208

24	**48**	**101**	**99**	**~248**
32	64	133	131	~328

Operationen pro Block (128 Samples)

M	sin() / Block	Mult/Block	Add/Block
8	2,047	4,737	4,481
16	4,095	8,832	8,576
24	**6,142**	**12,927**	**12,672**
32	8,190	17,022	16,768

Theoretische Komplexität pro Sample

```

sin() Aufrufe:      2 × M
Multiplikationen: ~6 + 4×M
Additionen:        ~4 + 4×M

```

5. Speicherbedarf

Für M=24 (empfohlene Konfiguration)

Komponente	Berechnung	Bytes
Filterkoeffizienten	$M \times 4$	96
Input-Buffer	128×4	512
Output-Buffer	128×4	512
State-Variablen	2×4	8
Gesamt		**1,128**

Für verschiedene M-Werte

M	Speicherbedarf [Bytes]
8	1,064
16	1,096

24	1,128
32	1,160

6. Frequenzselektivität

Der Filter ist inhärent frequenzselektiv

Der Block-LMS Filter verwendet ein Sinus-Referenzsignal bei exakt 50 Hz:

```
float sinReferenz = sinf(momentanePhase - k * phasenIncrement);
// wobei: phasenIncrement = 2pi × 50Hz / abtastFreq
```

Dämpfung bei verschiedenen Frequenzen (M=24, μ=0.01)

Frequenz [Hz]	Dämpfung [dB]	Bewertung
45	-11.6	Gedämpft
49	-16.4	Stark gedämpft
50	**-18.3**	**Stark gedämpft**
51	-21.5	Stark gedämpft
55	+5.3	Unverändert
100	+5.4	Unverändert
200+	~0	Unverändert

Notch-Charakteristik

Notch-Tiefe: ~18-21 dB bei 50 Hz

-3dB Breite: ca. 1.8 Hz (sehr schmalbandig)

Sprachsignal: Bleibt unverändert (Frequenzen > 100 Hz)

7. Stabilitätsanalyse

Instabile Kombinationen (zu vermeiden!)

M	Kritisches μ	Ergebnis
14	0.1	-32 dB (instabil)
16	0.1	-45 dB (instabil)
20	0.1	-50 dB (instabil)
24	0.1	-52 dB (instabil)
32	≥ 0.05	instabil

Stabilitätsbedingung

Für stabilen Betrieb sollte gelten:

$$\mu < 1 / (M \times E[x^2])$$

Empfehlung: Bei höherem M sollte μ kleiner gewählt werden.

8. Einschwingverhalten

Konvergenzzeit (bei M=24, $\mu=0.05$)

Erste Verbesserung: Nach ~20 ms (1 Periode der 50 Hz)

90% Konvergenz: Nach ~50 ms

Vollständige Konvergenz: Nach ~100 ms

Einflussfaktoren

Parameter	Einfluss auf Konvergenz
Höheres μ	Schneller, aber instabiler
Höheres M	Bessere Unterdrückung, langsamere Konvergenz
Stärkeres Signal	Schnellere Konvergenz
Stille	Adaptive μ -Reduktion verhindert Drift

9. Implementierungsdetails

Konfigurationsparameter (LMS.hpp)

```
#define SILENCE_THRESHOLD 1e-7f // Stille-Erkennung
#define LEAKAGE_FACTOR 0.999f // Koeffizienten-Decay bei Stille
#define MAX_COEFF_VALUE 10.0f // Clipping-Grenze
#define ENERGY_SMOOTHING 0.99f // Energieschätzung
```

Stabilitätsmechanismen

Adaptive μ -Reduktion bei Stille: ` $\mu \times 0.01$ ` wenn `signalEnergie < SILENCE_THRESHOLD`

Leakage bei Stille: Verhindert Koeffizienten-Drift

Clipping: Begrenzt Koeffizienten auf `[-10, +10]`

10. Empfehlungen für DSP-Board

Optimale Konfiguration

```
const int M = 24;
const float mu = 0.05f; // oder 0.01 für mehr Stabilität
const int BLOCK_SIZE = 128;
const float STOER_FREQ = 50.0f;
```

Trade-offs

Priorität	M	μ	Unterdrückung	Rechenaufwand
Qualität	32	0.01	20 dB	Hoch
Ausgewogen	**24**	**0.05**	**21 dB**	**Mittel**
Ressourcen	16	0.05	10 dB	Niedrig

Echtzeit-Anforderungen bei 8 kHz

Für M=24:

Samples pro Sekunde: 8,000

sin() Aufrufe/s: 384,000

Multiplikationen/s: 808,000

Zeit pro Block (128 Samples): 16 ms verfügbar

11. Fazit

Der Block-LMS Algorithmus eignet sich hervorragend zur adaptiven Unterdrückung von 50Hz Netzbrummen:

Frequenzselektiv: Nur 50 Hz wird unterdrückt, Sprache bleibt unverändert

Adaptiv: Passt sich automatisch an Amplitude und Phase der Störung an

Ressourcenschonend: Nur ~1.1 kB Speicher für M=24

Stabil: Mit richtiger μ -Wahl zuverlässiger Betrieb

Effektiv: Bis zu 21 dB Störunterdrückung möglich

Empfohlene Konfiguration für FM4-176-S6E2CC-ETH

Parameter	Wert
M	24
μ	0.05 (oder 0.01 für kritische Anwendungen)
Blockgröße	128 Samples
Speicher	~1.1 kB
Rechenaufwand	~248 Operationen/Sample

Erstellt mit Python-Testskripten: test_blockLMS.py, test_blockLMS_kontinuierlich.py, analyse_frequenzselektivitaet.py