

Persistente modulare Phasenzustände für die GFSK-Wellenformerzeugung

Eine reproduzierbare WSJT-X-Fallstudie

Rudolf Stepan

Unabhängiger Forscher, Wien, Österreich

OE3SRO | ORCID: 0009-0004-2842-2579

15. Oktober 2026 | Journal-Forschungsentwurf, Version 0.1

Vollständige deutsche Fassung des englischen Journalentwurfs

**Nicht begutachtet; menschliche fachliche und redaktionelle Überarbeitung
erforderlich.**

Zusammenfassung

Das Erhalten eines begrenzten modularen Phasenzustands kann in rekursiven Oszillatoren eine allgemeine Argumentreduktion vermeiden. Wir untersuchen dieses etablierte Entwurfsprinzip in den Generatoren für gaußgeformte Frequenzumtastung eines implementierten WSJT-X-Forks. Stepans Manuskript zur REIST-Division motiviert die Formulierung als persistenten zentrierten Zustand; die vorliegende Studie bewertet dessen Gleitkommaanwendung, ohne eine neue Methode der modularen Arithmetik zu beanspruchen. Zu den kontrollierten Alternativen zählen ein positives Phasenintervall, eine Maskenkorrektur auf Quelltextebene, eine allgemeine Restfunktion, eine 48-Bit-Festkomma-DDS, ein unbeschränkter Akkumulator und zwei Blockpräfix-Verfahren. Die produktive Implementierung bleibt unverändert. Auf einem Intel Core i9-14900K zeigen je drei Prozesssitzungen pro verifizierter Kernklasse eine wesentliche Abhängigkeit der Architekturwahl von der Ausführungssequenz: Das Medianverhältnis von allgemeiner Restfunktion zu zentrierter Korrektur beträgt auf dem ausgewählten Performance-Kern ungefähr 2,6 und auf dem ausgewählten Efficiency-Kern 4,4–5,2. Positive und zentrierte begrenzte Zustände verursachen ähnliche Kosten. Die geprüften Festkomma- und Präfiximplementierungen sind langsamer, ohne damit eine allgemeine Rangfolge von DDS-Architekturen zu begründen. Numerische Prüfungen umfassen gültig codierte Frames, lange Zustandsfolgen, Randfälle und Softwarespektren. Getrennte LDPC-Experimente zeigen unter bestimmten Bedingungen Verbesserungen des Verhältnisses von Erfolg zu Kosten gegenüber alleiniger BP-Decodierung; eine adaptive Costas-Suche verliert dagegen gültige Spitzen. Diese Decoderbefunde sind unabhängig von REIST und belegen keine verbesserte Funkempfindlichkeit. Rohdaten, exakte Quelltexthashes, automatisierte Auswertung und ausdrücklich benannte Einschränkungen begleiten den Entwurf.

Schlüsselwörter: modulare Phasenakkumulation; GFSK; DDS; reproduzierbare Leistungsmessung; Gleitkommaarithmetik; WSJT-X.

1 Einleitung

Ein Oszillator kann seine Phase als wachsenden Winkel oder als persistenten modularen Zustand darstellen. Beide Darstellungen sind mathematisch gültig, unterscheiden sich jedoch hinsichtlich Rechenaufwand, Rundung und Integrationsanforderungen. In einem schmalbandigen GFSK-Generator hängt jedes Sample von der vorherigen Phase und einem glatt veränderlichen Frequenzinkrement ab. Der Ersatz einer allgemeinen Restfunktion durch ein begründetes begrenztes Update kann diese Abhängigkeitskette verkürzen. Daraus folgt nicht, dass eine andere Station die gesendeten Signale leichter empfängt.

Stepans Manuskript zur REIST-Division [1] formuliert zentrierte ganzzahlige Reste als persistente Zustände mit einer begrenzten additiven Korrektur. Sein Abschnitt 6 behandelt ausdrücklich Phasenfenster. Unsere Anwendung verwendet dieselbe Idee der Zustandsführung mit binärer Gleitkommaphase und einer festen dargestellten Periode. Die elementare Invariante wird nachfolgend unabhängig bewiesen. Die CPU- und FPGA-Beschleunigungsfaktoren der ursprünglichen Arbeit werden nicht als Nachweise für diese Anwendung übernommen.

Der historische deutschsprachige Anwendungsbericht [2] verglich zentrierte Korrektur und allgemeine Restfunktion bei kontrollierter Strategie der trigonometrischen Auswertung. Er vermaß weder positive Zustände noch Festkomma-DDS und trennte Performance- und Efficiency-Kerne nicht. Die vorliegende Erweiterung untersucht:

1. Unter welchen arithmetischen Voraussetzungen erhält das begrenzte Update einen kanonischen Phasenzustand, und welche Fehler verbleiben?
2. Wie unterscheiden sich alternative Phasenarchitekturen auf diesem heterogenen Prozessor bei identischen Bedingungen für Pulse, Frames, Ausgabe und Compiler?
3. Welcher Anteil eines Gewinns im Wellenformkern kann für einen vollständigen Dateidecoder relevant werden?
4. Können getrennte Synchronisations- und LDPC-Experimente den Erfolg pro gemessenem Rechenaufwand erhöhen, ohne Verluste oder falsche Annahmen zu verdecken?

Der Beitrag besteht in einer durch Quelltextprüfung abgesicherten Implementierungsfallstudie und einem reproduzierbaren Vergleich einschließlich negativer Ergebnisse. Wir beanspruchen weder die Priorität für modulare Oszillatoren noch eine universelle Überlegenheit zentrierter Arithmetik.

2 Verwandte Arbeiten und Beitragszuordnung

Konventionelle DDS verwendet ein akkumuliertes Phasenregister und eine Amplitudenabbildung; Frequenzauflösung und Phasenabschneidung sind bekannte Entwurfsaspekte etablierter technischer Darstellungen [3]. Ein Gleitkommazustand in $[0, T)$ ist deshalb ein notwendiger Vergleichsfall und nicht grundsätzlich eine unterlegene Darstellung. Eine allgemeine Restfunktion unterstützt größere Definitionsbereiche und eine festgelegte Behandlung von Mittelpunktswerten; ein begrenztes Update nutzt dagegen zusätzliche Eingangsinvarianten.

Gleitkommaarithmetik unterscheidet exakte Identitäten von gerundeten Implementierungen. Auslöschung und Bedingungen für exakte Subtraktion sind für das begrenzte Update relevant [4]. Präfix-Scans sind etablierte parallele Grundbausteine [5]; ihre Anwendung auf die Phase beseitigt einen Teil der rekursiven Arbeit, ordnet Additionen jedoch um und verursacht zusätzlichen Rechen- und Speicheraufwand.

FT4 und FT8 kombinieren GFSK, Synchronisationsmuster und Fehlerkorrektur, wie von ihren Entwicklern dokumentiert [6]. Geschichtete LDPC-Ablaufpläne und Min-Sum-Näherungen sind Stand der Technik; beispielsweise untersuchen Cui et al. [7] eine spaltenweise geschichtete Hardwaredecodierung. Unser experimenteller zeilenweise geschichteter Softwarekern ist nicht deren Architektur und keine neue Entdeckung der geschichteten Decodierung.

Die bereitgestellte REIST-PDF nennt Rudolf Stepan, den angegebenen Titel, das Jahr 2026, ORCID und DOI [1]. Ihr SHA256 ist im Audit dokumentiert. Öffentliche DOI- und Zenodo-Metadaten konnten in dieser Untersuchung nicht abgerufen werden. Die DOI wird deshalb ausdrücklich als Kennung laut Autorenmanuskript und nicht als unabhängig verifizierte Metadatenangabe

Tabelle 1: Bekannte Verfahren und ihre tatsächliche Beziehung zu dieser Studie.

Verfahren/Quelle	Prinzip	Unterschied und empirische Rolle
Persistenter zentrierter Zustand, Stepan [1]	Ganzzahliger Zustand mit begrenzter vorzeichenbehafteter Korrektur	Motiviert die Anwendung. Gleitkommainvarianten und GFSK-Zielkonflikte werden hier untersucht; frühere Benchmarkzahlen werden nicht übernommen.
DDS/NCO [3]	Modulares Phasenregister, Amplitudenabbildung	Positiver Gleitkommazustand und Q48-Ganzzahlzustand sind konventionelle konkurrierende Implementierungen.
Allgemeine Restfunktion	Reduktion eines umfassend gültigen Eingabewerts bei jedem Update	Dieselben Double-Inkrement und dieselbe LUT; gemessene Referenz mit breiterem Funktionsumfang.
Präfix-Scan [5]	Umordnung von Teilsummen	Serielle und baumförmige Block-64-Architekturen; zusätzliche Reduktionen und Rundungen sind eingeschlossen.
Hybride FT-Decodierung [6]	Iterative BP und OSD	Ausgangspunkt einer getrennten Decoderstudie; kein Beitrag zur modularen Phase oder zu REIST.

behandelt. Eine andere DOI in Referenz [3] dieses Manuskripts wurde einem abweichenden Verlagsartikel zugeordnet. Diese Referenz wird hier nicht übernommen. Die Anwendungsbeziehung stützt sich auf die Definitionen und die additive Invariante des bereitgestellten Manuskripts, nicht auf dessen fehlerhafte Literaturangabe.

3 Mathematische Grundlagen

3.1 Zustandsinvariante über reellen Zahlen

Sei $T > 0$. Der kanonische zentrierte Repräsentant ist

$$C_T(x) = x - T \left\lfloor x/T + \frac{1}{2} \right\rfloor \in [-T/2, T/2). \quad (1)$$

Für $r \in [-T/2, T/2)$ und $u \in [-T/2, T/2]$ liegt die Summe $s = r + u$ in $[-T, T)$. Folglich ist

$$r^+ = \begin{cases} s - T & s \geq T/2, \\ s + T & s < -T/2, \\ s & \text{sonst} \end{cases} \quad (2)$$

zu s kongruent und liegt im kanonischen Intervall. Die drei Fälle werden auf $[-T/2, 0)$, $[0, T/2)$ beziehungsweise das unveränderte Innere abgebildet. Induktion begründet die persistente Erhaltung. Dies ist die Anwendung des additiven Zustandsprinzips aus der ursprünglichen REIST-Formulierung [1, Satz 3.6 und Korollar 3.7].

Für einen positiven Zustand $p \in [0, T)$ mit derselben vorzeichenbehafteten Schrittbegrenzung liegt $p + u$ in $[-T/2, 3T/2)$. Eine Addition bei negativer Summe oder eine Subtraktion bei einer Summe von mindestens T genügt ebenfalls. Der Verzicht auf allgemeine Division folgt somit aus dem begrenzten Zustand und begrenzten Inkrementen und nicht ausschließlich aus der Zentrierung.

3.2 Binäre Gleitkommainvariante und Fehler

Wir schreiben $P = \text{double}(\pi)$ und $\hat{T} = 2P$; die Multiplikation mit zwei ist exakt. Vorausgesetzt werden binary64, Rundung auf den nächsten darstellbaren Wert, endliche gespeicherte Werte

$r \in [-P, P)$ und $u \in [-P, P]$ sowie keine Umordnung, welche die angegebenen Operationen verändert. Aus der Monotonie der Rundung folgt $s = \text{fl}(r + u) \in [-2P, 2P]$; der gerundete obere Endpunkt kann erreicht werden. Bei $s \geq P$ ist die Subtraktion von $2P$ aufgrund der Bedingung für exakte Subtraktion bei einem Betragsverhältnis von höchstens zwei exakt (Sterbenz-Lemma). Bei $s < -P$ entspricht die Addition von $2P$ der entsprechenden exakten Subtraktion vergleichbar großer positiver Beträge. Diese Fälle liefern $[-P, 0]$ beziehungsweise $[0, P)$; das Innere liegt bereits in $[-P, P)$. Die binäre Zustandsinvariante gilt daher unter diesen ausdrücklich genannten Bedingungen, einschließlich einer gerundeten Summe von $2P$. Dies ist eine stärkere Aussage als eine Prüfung ausschließlich reeller Arithmetik, aber kein Beweis für NaNs, geänderte Rundungsmodi, Fast-Math-Transformationen oder beliebige Eingaben einer öffentlichen API.

Für normale Summen tritt die einzige lokale Rundung des Zustandsupdates bei der ersten Addition auf. Ihr absoluter Fehler ist durch $\epsilon_u \hat{T}$ mit $\epsilon_u = 2^{-53}$ begrenzt. Bei graduellem Unterlauf kann ein zusätzlicher subnormaler Term berücksichtigt werden. Eine konservative Schranke des akkumulierten Kreisabstands für n bereits dargestellte Schritte lautet

$$d_{\hat{T}}(\hat{r}_n, r_n) \leq n\epsilon_u \hat{T}, \quad d_T(a, b) = \min_{k \in \mathbb{Z}} |a - b - kT|, \quad (3)$$

bis auf subnormale Terme. Dies ist eine Worst-Case-Schranke und keine Driftvorhersage. Fehler bei der Bildung der Inkremente und der Darstellung des tatsächlichen 2π sowie mögliche Fehler des Restfunktions-Schutzpfads für große Schritte sind darin nicht enthalten.

Die allgemeine Funktion `std::remainder` verwendet eine Nächste-Ganzzahl-Konvention für Mittelpunktswerte und kann bei exakt einer halben Periode $+P$ liefern, anders als der kanonische zentrierte Zustand. Ein einfaches positives Update kann außerdem den gespeicherten Wert $\hat{T}$ erzeugen, wenn die Addition von $\hat{T}$ zu einem winzigen negativen subnormalen Wert aufgerundet wird. Diese Zustände bleiben periodisch sinnvoll; die Zugehörigkeit zum kanonischen Intervall und die Äquivalenz des Oszillators dürfen nicht verwechselt werden. Eigene Randfalldaten dokumentieren beide Fälle.

3.3 Festkomma und Umordnung

Der experimentelle Q48-Zustand stellt Periodenanteile durch $q/2^{48}$ dar. Sein Update lautet

$$q^+ = (q + \text{round}(2^{48}u/\hat{T})) \bmod 2^{48}. \quad (4)$$

Vorzeichenlose Addition und Maskierung sind exakt. Bei exakter Umrechnung trägt die Schrittquantisierung höchstens $n\hat{T}/2^{49}$ zum Kreisfehler bei; die tatsächliche Double-Skalierung und Rundung sind zusätzlich zu berücksichtigen. Kleine Schritte können auf null quantisiert werden. Diese Implementierung rechnet jeden veränderlichen GFSK-Schritt mit `llround` um; sie beansprucht nicht den bestmöglichen DDS-Umrechnungspfad.

In einem Block von $b = 64$ Samples sind die Versätze Präfixsummen der Inkremente. Sie können mehrere Perioden umfassen, weshalb eine einzige begrenzte Korrektur pro Versatz nicht ausreicht. Das Experiment verwendet eine Floor-basierte Reduktion der ausgegebenen Versätze und eine Restfunktion für den Übertragszustand des Blocks. Serielle Präfixakkumulation behält eine kurze Additions-kette bei; ein Baum-Scan stellt unabhängige Arbeit auf Kosten zusätzlicher Arithmetik und zusätzlichen Speicheraufwands bereit. Beide Varianten ändern die Rundungsreihenfolge. Der Baum-Arbeitsbereich aus zwei Arrays belegt 1024 Byte für Double-Werte, zusätzlich zu den unveränderten Wellenform-Arbeitspuffern.

4 Algorithmischer Entwurf und Implementierung in WSJT-X

Der produktive Generator verwendet Double-Inkremente und eine zentrierte Double-Phase. Anschließend interpoliert er linear in einer komplexen Float-Tabelle mit 65 536 Intervallen. Die ideale lineare Interpolationsschranke $h^2/8$ mit $h = 2\pi/65536$ beträgt ungefähr $1,15 \cdot 10^{-9}$ je Komponente; Float-Tabellenspeicherung und gerundete Interpolation sind getrennte Fehlerquellen. Die Phase wird vor dem Update abgetastet. Rampen und Bildung des Gaussian-Pulses bleiben modusspezifisch und unverändert.

Tabelle 2: Frameparameter bei 48 ksampl/s in diesem Fork.

	FT8	FT4	FT2
Modulationstöne	8	4	4
Eingabesymbole	79	103	103
Samples/Symbol	7680	2304	1152
Ausgabesamples	606720	241920	120960
Dauer (s)	12,64	5,04	2,52
Gaussian-Parameter BT	2	1	1

Der Code wird durch Commit

46ebbd78c663bf07eed054570df4888a6268e156

und die neuen Experiment-Dateihashes identifiziert. Jeder der 45 Dateihashes des historischen Manifests stimmt mit diesem Commit überein. Die historische Beschreibung eines noch nicht committeten Schnapsschusses war zum damaligen Zeitpunkt korrekt; das ursprüngliche Manifest bleibt erhalten und wird nicht rückwirkend umgeschrieben.

Tabelle 3: Implementierungsstellen; Pfade sind relativ zum Repository-Hauptordner.

Datei/Funktion	Aufgabe
<code>lib/ft8/native/ft8_wave.cpp</code> , <code>waveform</code>	Gemeinsame Phase, LUT, Pulse, Schutzpfade und Rampen für FT8/FT4/FT2.
Dieselbe Datei, <code>pulseFor</code>	Höchstens acht unveränderliche gemeinsam genutzte Pulsobjekte; eine Cache-Sperre pro Frame; Arbeitsdaten gehören jedem einzelnen Aufruf.
<code>journal/experiment/phase.hpp</code> , <code>oscillate</code>	Getrennte alternative Oszillatoren; generierte Kopien ersetzen ausschließlich den Oszillatorabschnitt.
<code>lib/ft4/native/ft4_sync.cpp</code> , <code>search</code>	Wiederverwendung frequenzkorrigierter Costas-Referenzen; unabhängige Zeilen können parallel laufen.
<code>lib/ft8/native/ft8_ldpc.cpp</code>	Produktiver nativer Encoder und zwei BP-Paritätsanordnungen; in dieser Studie unverändert.
<code>lib/ft8/decode174_91.f90</code>	Standard-Hybrid aus BP und OSD, auch von FT4/FT2 aufgerufen.
<code>tests/ft8/research/layered.cpp</code>	Experimentelle zeilenweise geschichtete SPA/Min-Sum; private Daten pro Aufruf, nicht Bestandteil der Produktion.

`journal/experiment` steht oben verkürzt für `docs/papers/reist-ftx/journal/experiment`. Die ursprünglichen Fortran-Wellenformquellen bleiben für Referenztests verfügbar. Unterschiede in Präzision, Trigonometrie und Cacheverwendung schließen eine Interpretation als Sprachleistungsvergleich aus. FT2 verwendet die FT4-basierte Integration des Forks und ist kein

weiterer unabhängiger LDPC-Code. Native C++-Experimente bedeuten nicht, dass verbleibende Fortran-Routinen interpretiert würden: Auch sie laufen bereits als kompilierter Maschinencode.

5 Experimentelle Methodik

5.1 Architekturvergleich

Das Testsystem ist ein Intel Core i9-14900K mit 24 physischen Kernen und 32 logischen CPUs unter Windows 11 Enterprise, Build 26300. Der Werkzeugsatz ist MSYS2 GNU GCC/GFortran 15.2.0. Die Wellenformexperimente verwenden C++11 und `-O3 -fno-fast-math -ffp-contract=off`, ohne LTO und ohne `-march=native`. Eine getrennte Prüfung mit Boost 1.90 und 100-stelliger Dezimalpräzision benötigt C++14; diese Anforderung an den Sprachstandard verändert die zeitlich vermessenen Kerne nicht.

Kernbindung und CPUID-Blatt 1A pro logischer CPU identifizieren Kerne des Typs 0x40 und 0x20 [8]. Der ausgewählte Performance-Thread ist logische CPU 0, der ausgewählte Efficiency-Thread CPU 16. SMT-Geschwister, Turbo, Energieverwaltung und Hintergrundlast sind nicht isoliert. Drei zusätzliche Sitzungen erlauben eine freie Ausführungszuordnung durch das Betriebssystem. Die Ergebnisse gelten für diese ausgewählten Kerne und nicht für jeden P-/E-Kern.

Für jede Kernklasse gibt es drei Prozesssitzungen, neun gepaarte randomisierte Blöcke je Modus und zehn Implementierungen. Ein Block umfasst nach dem Aufwärmen drei vollständige Frames je Implementierung. Die neue zeitlich vermessene Ausgabe besteht aus komplexen Float-IQ-Samples. Alle Varianten erhalten dasselbe codierte Telemetrierwort (Seed 991), 48 ksample/s und 1855 Hz. Die Telemetriebits werden mit dem vorhandenen LDPC-/CRC-Verfahren und den passenden Costas-, Gray- und FT4-Scrambling-Abbildungen codiert. Pulsbildung, Allokation, Cacheverwendung, Rampen und Ausgabegröße sind eingeschlossen. Der umstrukturierte zentrierte Oszillator muss im numerischen Testkorpus mit den Ausgabebytes der Produktion übereinstimmen.

Der Hauptschätzer ist ein Verhältnis gemeinsam ausgewerteter Mediane. Bedingte Unsicherheitsintervalle ziehen ganze Prozesssitzungen und darin gepaarte zirkuläre Blöcke der Länge drei erneut (2000 Replikate, Seed 991202610). Längen eins und fünf dienen als Empfindlichkeitsprüfungen. Das Resampling erhält lokale Zeitabhängigkeiten; bei lediglich drei Sitzungen ist die Überdeckungswahrscheinlichkeit nicht nachgewiesen. Gemeinsame thermische und Betriebssystemeffekte können auch getrennte Prozesse korrelieren. Es handelt sich um bedingte Resampling-Intervalle und nicht um garantierte Populations-Konfidenzintervalle oder plattformübergreifende Inferenz. Einzelne wiederholte Frames gelten nicht als unabhängige Stichproben.

Die verstrichene Zeit wird mit `steady_clock` gemessen. Windows `QueryThreadCycleTime` liefert als Diagnosegröße verbuchte Threadzyklen für Benutzer- und Kernelmodus, normiert pro Sample [9]. Dies sind weder Zählungen abgeschlossener Instruktionen noch kalibrierte PMU-Messungen. Hardware-Fehlvorhersagen von Sprüngen, Cache-Ereignisse, Energie und dynamische Instruktionen pro Sample wurden nicht erfasst. Compiler-Assembler für kleine Updateschleifen liegt bei; eine Maskenform auf Quelltextebene wird nicht in jedem Kontext als Garantie für sprungfreien Maschinencode betrachtet.

5.2 Historische Reproduktion, numerischer Korpus und Spektren

Der historische Harness mit reeller Ausgabe wird mit seinem ursprünglichen Arbeitsfall erneut in neue Dateien ausgeführt: 21 Blöcke, zehn Wiederholungen und Tonindizes mit Seed 991. Die zufälligen gültigen Tonindizes müssen keine Nachricht codieren. Diese Daten bleiben vom neuen Architekturvergleich mit komplexer Ausgabe und von den eingefrorenen Originalmessungen getrennt.

Der numerische Architekturkorpus enthält drei Modi, zwei Abtastraten (12/48 ksample/s), zwei codierte Telemetrienachrichten und fünf Träger pro Abtastrate: -1855 , $0,001$, 1855 , $0,49f_s$ und 10^9 Hz. Damit umfasst er 60 Framekonfigurationen und zehn Implementierungen. Der große Träger beansprucht denselben Restfunktions-Schutzpfad für Schritte und stellt keinen realistischen Funkbetrieb dar. Die Long-Double-Referenz akkumuliert dieselben dargestellten Double-Inkrementen modulo derselben Double-Periode und wertet Sinus/Kosinus in Long Double aus. Ihre 64-Bit-Mantisse bildet eine Referenz mit höherer arithmetischer Genauigkeit, aber keine exakte Referenz für Gaussian-Pulsbildung oder die physikalische Zahl Pi. Berechnungen mit 100 Dezimalstellen prüfen unabhängig 4096 lokale Updates; eine Million zufälliger Zustands-/Schrittpaare prüft die Intervallzugehörigkeit.

Getrennte Zustandsfolgen mit jeweils einer Million Schritten verwenden einen konstanten positiven Trägerschritt, vorzeichenbehaftete Zufallsschritte und winzige alternierende Schritte. Periodischer Phasenfehler, maximaler und RMS-IQ-Fehler sowie tatsächliche Bytegleichheit sind unterschiedliche Ausgabegrößen. Die Softwarespektren verwenden den vollständigen Frame einschließlich Rampen, ein rechteckiges Beobachtungsfenster, Nullauffüllung auf die nächste Zweierpotenz und eine deterministische Radix-2-FFT. Eine Kontrolle mit bekannter Frequenzbinlage und ein Parseval-Fehlermaß validieren die Transformation. Das gewählte Analyseband erstreckt sich um zwei Tonabstände über die nominell niedrigsten/höchsten Töne hinaus; dies ist keine regulatorische Emissionsmaske. Die Fehlerenergie wird auf die Energie des Referenzframes normiert. Sämtliche Größen betreffen erzeugte digitale IQ-Samples.

5.3 Komponentenprofilierung im vollständigen Decoder

Ein ausdrücklich vom Standardbuild ausgeschlossenes Programm bindet den unveränderten produktiven Wellenformquelltext ein und misst dessen ABI-Aufrufe innerhalb des vollständigen `jt9`-Dateidecoders. FT8-, FT4- und optionale lokale FT2-Aufnahmen werden mit Decodiertiefe drei, einem OpenMP-Thread und drei gemessenen Prozessen nach einem Aufwärmloop verarbeitet. Die ausgegebenen Nachrichtenmengen müssen zwischen den Läufen identisch bleiben. Der ausgewiesene Komponentenanteil teilt die aufsummierte Wellenformzeit durch die vom Prozessstart bis zum Ende verstrichene Zeit; er enthält den Programmstart und gilt für diese Dateien. Es handelt sich nicht um eine Profilierung der GUI, eines Livestreams oder des Transceivers.

5.4 Unabhängige Decoder- und Synchronisationsstudien

Der erste Decoder-Pilot verwendet 128 Wörter je Zelle, Seed 20261016. Eine getrennte Bestätigung verwendet 2048 Wörter je Zelle und Seed 20261017. Fünf Szenarien umfassen BPSK-AWGN-LLRs, einen Ausfall von zwölf aufeinanderfolgenden Bits mit LLR null, 28 korrekt eingefrorene AP-Bits, ein falsch eingefrorenes AP-Bit und ausschließlich Rauschen, jeweils bei $\sigma = 0,8; 1,0; 1,2$. Das Kanalmodell lautet $y = b + \sigma z$, $b \in \{-1, 1\}$ und $L = 2y/\sigma^2$; ein positives LLR steht für Bit eins. Sein $E_s/N_0 = 1/(2\sigma^2)$ ist eine BPSK-Kanalgröße und *kein* FT-Modus-SNR in 2500 Hz Bandbreite. Nutzdaten und Rauschen unterscheiden sich zwischen den Versuchen, stimmen aber zwischen den Verfahren überein. Box-Muller und eine explizite Umrechnung von `mt19937` auf Gleichverteilung vermeiden bibliotheksspezifische Transformationen von Normalverteilungen.

Beide Anordnungen der Paritätszeilen werden geprüft. Vergleichsbasis sind native Flooding-BP30 und für die Standardanordnung der bestehende Hybrid mit zwei gespeicherten BP-Zuverlässigkeitssummen und OSD-Ordnung zwei. Alternativen sind zeilenweise geschichtetes Sum-Product (15 Durchläufe) und normalisiertes Min-Sum (30 Durchläufe mit $\alpha = 0,75$ oder $0,85$ sowie 60 Durchläufe mit $0,75$). SPA-Prüfknottennachrichten werden auf ungefähr ± 14 begrenzt. Der geschichtete Zustand wird unmittelbar nach jeder Prüfzeile aktualisiert. AP-Werte bleiben eingefroren.

Nach der Bestätigung werden zwei durch die Daten motivierte Kombinationen mit günstigem Verlauf festgelegt: MS085, bei Fehlschlag gefolgt von unverändertem BP30 oder dem Standardhybrid. Sie verwenden einen neuen Datensatz mit 2048 Wörtern je Zelle, Seed 20261018, ohne ihn mit den Bestätigungsdaten zusammenzuführen. Ein fälschlich angenommenes Wort kann den Rückfall verhindern; die Erhaltung des bisherigen Erfolgs ist deshalb ein empirisches Ergebnis und keine universelle Aussage. Die SMP-spezifische OSD wird nicht durch Einsetzen des Standardhybrids als vermessen betrachtet.

Jeder Erfolg muss Syndrom- und CRC-Prüfungen erfüllen und mit dem bekannten vollständigen 174-Bit-Sollwort übereinstimmen. Falsche Annahmen einschließlich Nullwörtern bleiben erfasst, auch wenn die Anwendung weitere semantische Filter anwenden würde. Gepaarte Gewinne/Verluste werden ausgewiesen. Marginale Wilson-Intervalle beschreiben jeden Korrektheitsanteil; sie sind keine Intervalle für gepaarte Differenzen. Die Zeitmessungen umfassen serielle Decoderaufrufe in randomisierten 32-Wort-Blöcken und schließen Datenerzeugung und CSV-Ein-/Ausgabe aus. Dies ist ein Vergleich mittlerer Kosten und keine Garantie einer festen Zeitgrenze je Aufruf.

Das Costas-Experiment injiziert vier Referenzarrays in komplexes Rauschen und vergleicht das bestehende vollständige Raster (585 Punkte) mit einem groben 85-Punkte-Raster und einer kleinen lokalen Verfeinerung. Es verwendet 1024 Versuche je Standardabweichung des Real-/Imaginär rauschens 1, 2, 4, 8, Seed 174202610, zufällige Zeit- und Frequenzlagen sowie einen Thread. Eine Lokalisierung gilt innerhalb von zwei heruntergetasteten Samples und einem Hz als korrekt. Gemessen wird ein vereinfachter Ersatzversuch für den Suchkern, nicht vollständige Framedecodierung, FT8-Suchtrefferquote, Interferenzsubtraktion oder eine gemessene RF-SNR-Kurve.

6 Leistungsergebnisse

Tabelle 4: Wellenformzeit mit komplexer Ausgabe auf dem ausgewählten P-Kern: Median von 27 Blockmittelwerten, Millisekunden pro vollständigem Frame. Die gesamte Puls- und Rampenarbeit ist eingeschlossen.

Architektur	FT8 (ms)	FT4 (ms)	FT2 (ms)
Zentriert	2,774	1,088	0,550
Restfunktion	7,201	2,857	1,431
Positiv	2,860	1,089	0,601
Maskenform	2,876	1,134	0,557
Q48-DDS	6,482	2,672	1,332
Unbeschränkt	10,857	4,347	2,227
Seriell Prefix-64	4,758	1,890	1,024
Baumpräfix-64	5,864	2,403	1,260
Long double	27,437	10,588	5,386
Produktion	2,719	1,038	0,528

Positive begrenzte Zustände liegen in diesen Messungen nahe an zentrierten Zuständen. Die praktische Einsparung entsteht hauptsächlich durch Vermeiden wiederholter allgemeiner Reduktion. Eine Maskenform auf Quelltextebene liefert keine konsistente Verbesserung. Die geprüfte Q48-Umrechnung und die Block-Scans verursachen mehr Kosten als die skalare zentrierte Korrektur. Dies sind negative Ergebnisse für die konkreten Implementierungen und keine Beweise, dass optimierte Ganzzahl-DDS oder explizites SIMD nicht gewinnen könnten.

Die Wiederholung prüft die Wiederholbarkeit unter dem aktuellen unkontrollierten Systemzustand und nicht die exakte Gleichheit verstrichener Zeiten. Unterschiede zwischen P-/E-Klassen

Tabelle 5: Kernabhängigkeit der allgemeinen Restfunktion (G) gegenüber zentrierter Korrektur (C). Die Intervalle sind bedingte hierarchische Resampling-Intervalle; siehe Methodik.

Kern	Modus	G/C	Bedingtes 95%-Intervall	C (ns/Sample)
P	FT8	2,60	[2,35; 2,71]	4,57
P	FT4	2,62	[2,32; 2,75]	4,50
P	FT2	2,60	[2,53; 2,97]	4,55
E	FT8	5,19	[5,05; 5,33]	5,96
E	FT4	4,94	[4,70; 5,08]	6,37
E	FT2	4,43	[4,27; 4,61]	7,04
OHNE BINDUNG	FT8	2,63	[2,62; 2,67]	4,16
OHNE BINDUNG	FT4	5,03	[2,41; 5,17]	6,46
OHNE BINDUNG	FT2	5,53	[1,98; 5,85]	5,62

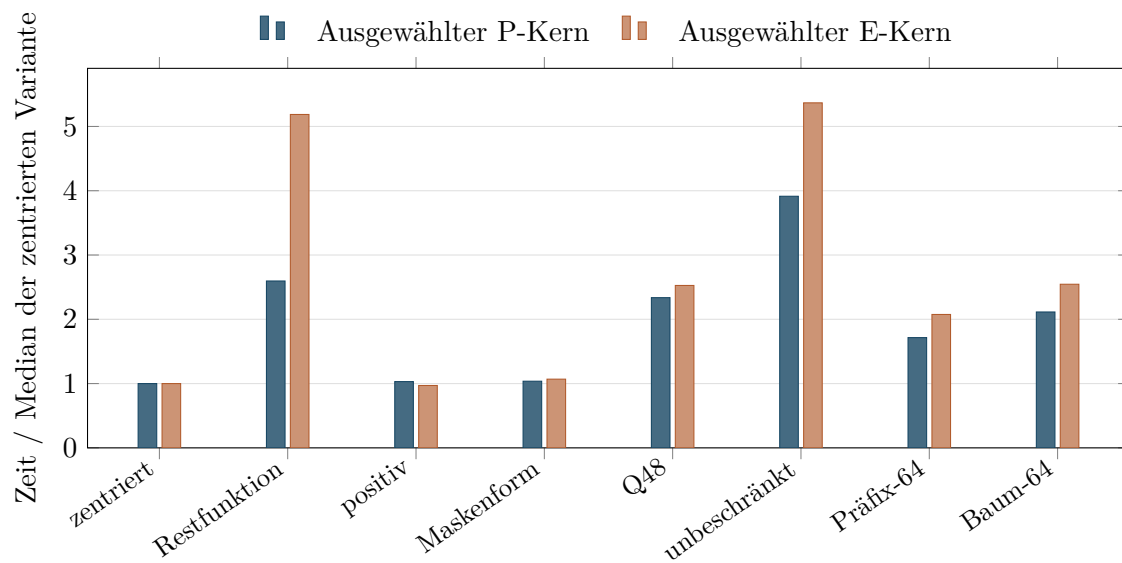


Abbildung 1: FT8-Architektorkosten, innerhalb jeder Kernklasse getrennt normiert. Die Abbildung zeigt Punktschätzer; Intervall- und Empfindlichkeitsdaten stehen in der maschinenlesbaren Zusammenfassung.

Tabelle 6: Getrennte Wiederholung des historischen faktoriellen Harness mit reeller Ausgabe. Diese Verhältnisse dürfen nicht mit dem neuen Vergleich komplexer Ausgabe gleichgesetzt werden.

Modus	Historisch G-L/C-L	Neue Reproduktion, reelle Ausgabe
FT8	5,55	5,56
FT4	5,24	5,21
FT2	4,99	5,19

und ungebundenen Prozessen zeigen, weshalb die früheren Faktoren 5,55/5,24/4,99 nicht zu maschinenunabhängigen Konstanten erklärt werden können. Compilerkontext und Ausgabetyt unterscheiden sich ebenfalls zwischen historischem und erweitertem Harness.

Parallelversuche verwenden unabhängige zentrierte Wellenformaufrufe mit 1/2/4/8 Workern, fünf Blöcken und acht Frames je Worker. Sie umfassen Threaderzeugung, Allokation der Arbeitsdaten und Cache-Sperren und schließen kerngebundene Sitzungen bewusst aus. Die vollständige rohe Durchsatztabelle bleibt erhalten. Es handelt sich um den Durchsatz unabhängiger Frames und nicht um die SIMD-Beschleunigung eines einzelnen rekursiven Oszillators oder einen Nachweis vergleichbarer Skalierung eines Decoders mit zwölf Workern.

7 Numerische Genauigkeit und Signalvalidierung

Tabelle 7: Schlechtestes Ergebnis über den 60-Frame-Korpus gegenüber der Long-Double-Referenz. IQ-Maximum und RMS verwenden den komplexen Betrag; Phasenwerte schließen Rampenbereiche mit Referenzamplitude von höchstens 0,1 aus.

Architektur	Max. IQ-Fehler	Max. RMS-Fehler	Max. Phasenfehler (rad)
Zentriert	8,43e-08	3,20e-08	1,30e-07
Restfunktion	8,43e-08	3,20e-08	1,30e-07
Positiv	8,43e-08	3,20e-08	1,30e-07
Maskenform	8,43e-08	3,20e-08	1,30e-07
Q48-DDS	8,43e-08	3,20e-08	1,30e-07
Unbeschränkt	2,22e-06	8,80e-07	2,27e-06
Seriell Prefix-64	8,43e-08	3,20e-08	1,30e-07
Baumpräfix-64	8,43e-08	3,20e-08	1,30e-07

Der umstrukturierte zentrierte Kern stimmt im gesamten Korpus bytegenau mit der Produktion überein. Andere Architekturen müssen für periodische Äquivalenz nicht bytegleich sein. Für eine Million konstanter positiver Schritte erreichen zentrierter Zustand und allgemeine Restfunktion ungefähr $6,86 \cdot 10^{-11}$ rad maximalen periodischen Fehler gegenüber der Long-Double-Referenz mit dargestellter Periode. Positive Akkumulation erreicht $1,53 \cdot 10^{-10}$ rad, Q48 ungefähr $2,38 \cdot 10^{-9}$ rad und der unbeschränkte Akkumulator ungefähr $4,40 \cdot 10^{-6}$ rad. Vorzeichenbehaftete Zufallsschritte verhalten sich anders; diese Werte dürfen nicht als universelle Driftraten interpretiert werden. Winzige positive Schritte können in Q48 vollständig verschwinden. Randfalltests zeigen den bereits beschriebenen Mittelpunktswert der allgemeinen Restfunktion und die Endpunkteffekte des positiven Zustands.

Die integrierte Fehlerenergie der zentrierten Variante gegenüber der Referenz beträgt bei den drei ausgewählten Frames ungefähr -150 dBc; die Q48-Implementierung liegt bei dieser durch die Float-Ausgabe begrenzten Fehlerhöhe ähnlich. Die relativen Parseval-Fehler liegen unter $7 \cdot 10^{-13}$. Die Referenzenergie außerhalb des ausdrücklich gewählten Analysebands beträgt für FT8/FT4/FT2 ungefähr $-38,1$, $-48,3$ und $-50,9$ dBc. Diese Werte hängen von Beobachtung, Rampe und gewähltem Band ab und stellen keine Zertifizierung einer Spektralmaske dar. Sie belegen keine verringerten RF-Seitenbänder. Die Long-Double-Referenz verwendet weiterhin dieselben dargestellten Pulsinkremente, dieselbe abschließende Float-Umwandlung und dieselben Rampenkoeffizienten wie die Varianten.

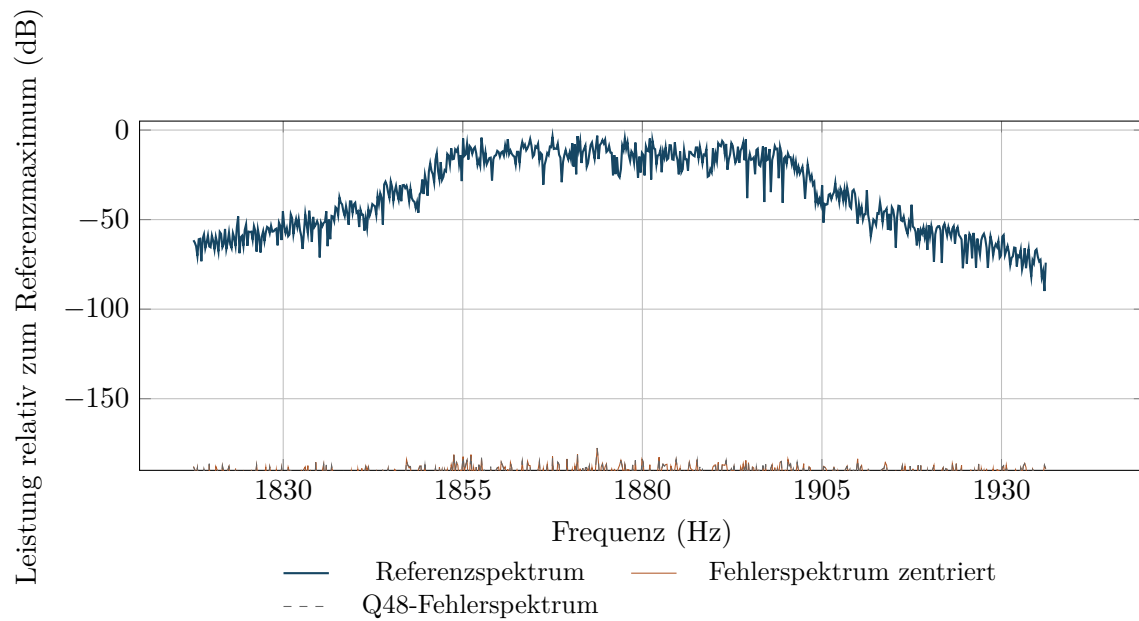


Abbildung 2: Ein erzeugter FT8-Telemetrie-Frame bei 1855 Hz und 48 ksample/s. Fehlerspektren beruhen auf IQ-Differenzen und nicht auf der Differenz logarithmischer Leistungskurven. Dies ist ein Softwarevergleich und keine Sendermessung.

Tabelle 8: Unveränderte Wellenformkomponente im vollständigen Einthread-Aufnahmedecoder; Mediane aus drei Läufen nach dem Aufwärmen. Die ideale Grenze entfernt diese Komponente vollständig und hält alle übrige Arbeit konstant.

Modus	Gesamt (ms)	Welle (ms)	Aufrufe	Anteil (%)	Ideale Grenze
FT8	1574,7	33,47	254	2,13	1,022
FT4	177,7	6,63	18	3,73	1,039
FT2	144,6	3,46	5	2,39	1,024

8 Praktische Relevanz und unabhängige Decoderergebnisse

Für den ursprünglichen Komponentenanteil p und die Komponentenbeschleunigung S lautet das Amdahl-Modell

$$S_{\text{total}} = \frac{1}{1 - p + p/S}, \quad S_{\text{total}} \leq \frac{1}{1 - p}. \quad (5)$$

Der hier gemessene Anteil gehört zur aktuellen zentrierten Implementierung. Um deren Ersatz durch eine langsamere allgemeine Restfunktion mit relativen Kosten G zu modellieren, lautet das entsprechende kontrafaktische Verhältnis $1 + (G - 1)p$. Dies ist ein bedingtes Modell; modusspezifische Abtastraten, Cachezustand und Eingabemischung können G verändern. Wir setzen kein 48-ksample/s-TX-Verhältnis an die Stelle eines exakt gemessenen RX-Verhältnisses. Die Profilierung zeigt, wo die aktuelle Komponente relevant ist, nicht einen gemessenen Gesamtprogrammgewinn durch hypothetischen Ersatz.

Tabelle 9: Getrennter Kombinationsdatensatz: Standard-Paritätsanordnung, BPSK-AWGN, 2048 unabhängige Wörter je Rauschstufe. Die Zeit ist der mittlere Blockaufwand relativ zu BP30. Falsche Annahmen zählen akzeptierte Wörter, die vom bekannten Sollwort abweichen.

Rauschen σ	Verfahren	Korrekt/2048	Gewinn/Verlust	Falsch	Zeit/BP
0,8	BP30	1377	0/0	0	1,00
0,8	Hybrid	1971	594/0	0	1,75
0,8	MS085	1499	132/10	0	0,13
0,8	MS→BP	1509	132/0	0	0,63
0,8	MS→Hybrid	1971	594/0	0	1,22
1,0	BP30	56	0/0	0	1,00
1,0	Hybrid	587	531/0	0	3,81
1,0	MS085	98	42/0	0	0,31
1,0	MS→BP	98	42/0	0	1,28
1,0	MS→Hybrid	589	533/0	0	4,02
1,2	BP30	0	0/0	0	1,00
1,2	Hybrid	31	31/0	0	6,61
1,2	MS085	3	3/0	0	0,59
1,2	MS→BP	3	3/0	0	1,58
1,2	MS→Hybrid	31	31/0	0	7,34

Bei $\sigma = 0,8$ liefert der neue Kombinationsdatensatz 1377 BP-Erfolge, 1499 MS085-Erfolge und 1509 Erfolge mit MS und anschließendem BP-Rückfall. MS085 gewinnt 132 Fälle und verliert zehn; die Kombination erhält diese zehn durch den Rückfall. Ihr gemessener mittlerer Aufwand beträgt ungefähr 0,63 des alleinigen BP-Aufwands. Damit wird die begrenzte Frage zum Kern für diese Verteilung bejaht: Eine höhere Wahrscheinlichkeit korrekter Wörter kann mit geringerem gemessenem mittlerem Budget einhergehen. Dies ist kein Nachweis einer festen Zeitgrenze oder eines vollständigen Empfängergerwinns. Bei $\sigma = 1,0$ und $1,2$ kostet dieselbe Kombination mehr als BP, weil die meisten Frames den Rückfall erreichen. Es besteht keine gleichmäßige Überlegenheit über alle Kanalbedingungen.

Der Standardhybrid bleibt wesentlich erfolgreicher als BP oder Min-Sum allein. Ein Min-Sum-Vorlauf hilft gelegentlich oder verringert seine Kosten, verursacht in schwachen beziehungsweise reinen Rauschfällen aber auch Zusatzaufwand. Die vollständigen Bestätigungs- und Kombinationsmatrizen enthalten AP-, Falsch-AP- und Rauschkontrollen, beide Paritätsanordnungen und alle geprüften Varianten. Beim ursprünglichen Hybrid tritt im Bestätigungsdatensatz für AWGN mit $\sigma = 1,2$ ein CRC-gültiges falsches Wort auf. Dies ist eine falsche Annahme des Kerns vor den semantischen Anwendungsfiltern und keine neu eingeführte falsche Funkmeldung. Null Beobachtungen in anderen endlichen Zellen beweisen keine Falschdecodiertrate von null.

Die adaptive Suche ist ungefähr vier- bis fünfmal schneller, verliert aber auch bei der stärksten

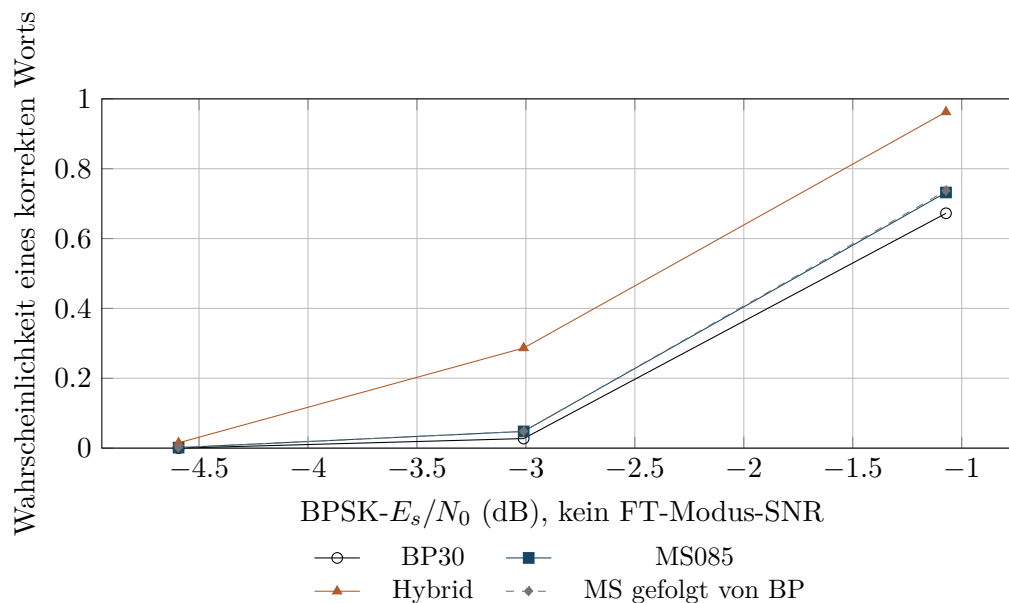


Abbildung 3: Drei untersuchte Kanalstufen im unabhängigen Kombinationsdatensatz. Die Linien dienen der Orientierung und sind keine angepassten Empfindlichkeitskurven. Wilson-Intervalle stehen in der begleitenden CSV-Datei und sind marginale Intervalle, keine gepaarten Tests.

Tabelle 10: Vereinfachte FT4-Costas-Lokalisierung: 1024 gepaarte Versuche je Rauschstufe. Die Zeittoleranz beträgt zwei heruntergetastete Samples, die Frequenztoleranz ein Hz.

Rauschen σ	Suche	Korrekt/1024	Gewinn/Verlust	Median (μ s)
1	vollständig	1024	0/0	122,4
1	adaptiv	1022	0/2	26,9
2	vollständig	1023	0/0	121,8
2	adaptiv	977	0/46	25,8
4	vollständig	524	0/0	122,5
4	adaptiv	440	46/130	27,0
8	vollständig	57	0/0	122,5
8	adaptiv	53	21/25	26,8

geprüften Signalstufe Spitzen. Bei Rauschstandardabweichung vier findet sie 440 korrekte Lokalisierungen gegenüber 524 beim vollständigen Raster, mit 46 Gewinnen und 130 Verlusten. Dies spricht gegen den produktiven Einsatz genau dieser Beschneidungsregel als Empfindlichkeitsverbesserung. Eine produktive Strategie müsste mehrere grobe Spitzen erhalten oder einen begründeten unsicherheitsabhängigen Rückfall verwenden und anschließend anhand vollständiger Nachrichten geprüft werden. Kein neuer Decoderalgorithmus dieser Studie wird in der Anwendung aktiviert.

9 Diskussion

Der Vergleich stützt einen persistenten begrenzten Zustand als nützliche Implementierungswahl für diesen GFSK-Kern. Er belegt keinen besonderen arithmetischen Vorteil der Zentrierung gegenüber jeder positiven Darstellung. Zentrierung liefert eine zweckmäßige vorzeichenbehaftete Fehlerkoordinate und unter den genannten binary64-Voraussetzungen exakte Korrektursubtraktionen. Die Größe des Leistungsvorteils gegenüber einer Bibliotheksrestfunktion hängt stark von Kern und Kontext ab.

Q48-DDS tauscht Rundungsfehler des Zustands gegen Schrittquantisierung und Umrechnungskosten. Die untersuchte Implementierung mit `llround` bei jedem Schritt ist kein optimierter Festkomma-Maßstab, dessen Überwindung allgemeine Überlegenheit belegt. Ebenso wenig bewertet ein arbeitsaufwendigerer Baum-Scan für die Standard-ISA sämtliche SIMD-Implementierungen. Explizite AVX2-Scans, zusammengeführte Puls-/Phasendarstellungen, vektorisierte Umrechnung und die gemeinsame Verarbeitung unabhängiger Oszillatoren bleiben plausible Folgestudien. Eine Umordnung erfordert eigene Fehlernachweise und sollte nicht durch Fast-Math-Optionen verdeckt werden.

Die getrennten Decoderergebnisse zeigen, weshalb Durchsatz und Empfindlichkeit gemeinsam gemessen werden müssen. Ein günstiger Vorlauf kann bei vorteilhaften Eingaben spätere Arbeit vermindern, bei überwiegend erfolglosen Fällen aber zusätzliche Zeit verbrauchen. Eine Suche, die die meisten Hypothesen verwirft, kann schneller sein und gleichzeitig echte Spitzen übersehen. Keines dieser Phänomene wird aus REIST oder aus der sendeseitigen Phaseninvariante hergeleitet.

10 Gefährdungen der Gültigkeit

Gemessen werden nur ein physisches System und eine Compilerversion. Es liegen keine AMD- oder ARM64-Ergebnisse, Hardware-PMU-Ereignisse, RF-Messungen oder Energiewerte vor. Die P-/E-Identität ist verifiziert; Kernisolation und Leistungskontrolle fehlen jedoch. Drei Prozesssitzungen und kurze Blöcke liefern begrenzte Informationen über thermische und Scheduler-Abhängigkeiten; bedingte Bootstrap-Intervalle können systematische Unsicherheit unterschätzen. Verhältnisse gemeinsam ausgewerteter Mediane unterscheiden sich von Medianen gepaarter Verhältnisse. Wiederholte Frames innerhalb eines Blocks sind keine unabhängigen experimentellen Einheiten.

Der erweiterte Oszillator-Harness verwendet gegenüber dem historischen faktoriellen Experiment mit reeller Ausgabe komplexe Ausgabe, andere codierte Eingaben und einen umstrukturierten Compilerkontext. Die historische Wiederholung wird daher getrennt ausgewiesen. Beide umfassen unveränderte Pulsbildung und Rampen; das gemessene Verhältnis betrifft somit die Kosten des gesamten Kerns und nicht einer isolierten Addition.

Long-Double-Referenzen verwenden dieselben Double-Inkrementen und dieselbe Periode; lokale Prüfungen mit 100 Dezimalstellen validieren nicht den gesamten Gaussian-Puls auf 100 Stellen. Der numerische Korpus ist endlich, enthält nur zwei Telemetrienachrichten je Einstellung und kann keine universelle Byteidentität beweisen. Die Spektren betreffen ausgewählte ideal erzeugte

Frames. Weder Audiointerface, USB-Pfad, Verstärker, Antenne noch FT-991A wurden vermessen. Die BPSK-LLRs des LDPC-Versuchs bilden FSK-Demodulation, Synchronisationsfehler, Fading-Diversität, Gleichkanalinterferenz und semantische Zurückweisung durch die Anwendung nicht ab. Zufällige 77-Bit-LDPC-Nutzdaten sind Nachrichten auf Codeebene und nicht ausnahmslos gültige angezeigte Nachrichtenformate. Die SMP-spezifische OSD-Strategie ist nicht mit der Standardhybrid-Vergleichsbasis gleichzusetzen. Die Costas-Injektion ist ein vereinfachtes Modell heruntergetasteter Referenzen ohne Gaussian-Übergänge oder vollständigen Decoder. Decoderwahrscheinlichkeiten können nicht in RF-Empfindlichkeit von FT8/FT4/FT2 oder eine Verbesserung in dB umgerechnet werden. Ein mittleres Zeitbudget belegt weder die Einhaltung von Zeitgrenzen je Frame noch den gesamten Mehrstationsdurchsatz.

Es liegen keine externe Replikation, unabhängige mathematische Prüfung oder Journalbegutachtung vor. Dieser Entwurf wurde wesentlich mit KI-Unterstützung erstellt. Sein Vorliegen begründet keine Einhaltung der Autoren- oder KI-Richtlinie eines bestimmten Journals. Der menschliche Autor muss jede spätere Einreichung kritisch überarbeiten und verantworten.

11 Schlussfolgerungen

Das Erhalten einer begrenzten modularen Phase vermeidet in der untersuchten GFSK-Implementierung unnötige allgemeine Reduktion. Eine präzise arithmetische Invariante erklärt, wann eine Korrektur genügt; tatsächlicher Fehler und Leistung hängen weiterhin von Darstellung, Rundung, Ausgabeabbildung und Architektur ab. Im erweiterten Vergleich ist ein positiver begrenzter Zustand mit dem zentrierten Zustand konkurrenzfähig; zugleich zeigen sich wesentliche P-/E-Kern-Effekte. Langsamere alternative Implementierungen und eingabeabhängige Decoder-Zielkonflikte werden ebenfalls dokumentiert, anstatt ausschließlich günstige Ergebnisse auszuwählen.

Die Beziehung zu REIST besteht in einer Anwendung seiner Formulierung als persistenter zentrierter Zustand, gestützt auf das bereitgestellte Manuskript und eine unabhängige Herleitung. Über den früheren Anwendungsbericht hinaus liefert diese Arbeit einen breiteren kontrollierten Vergleich, die Analyse arithmetischer Randfälle, kernspezifische Messungen und getrennte Decoderexperimente zum Verhältnis von Erfolg und Kosten. Eine weitergehende mathematische Priorität, eine RF-Verbesserung oder Einreichungsreife wird nicht beansprucht.

12 Verfügbarkeit von Daten und Code

Die Anwendungsbasis ist <https://github.com/rudolfstepan/wsjt-x>. Der Quellcommit lautet 46ebbd78c663bf07eed054570df4888a6268e156. Neue noch nicht committete Experimentdateien werden durch ein unveränderliches Quellmanifest im beigefügten Reproduktionspaket identifiziert. Eingefrorene historische Rohdaten verbleiben unter `docs/papers/reist-fts/data`; neue Daten liegen im getrennten Verzeichnis `docs/papers/reist-fts/journal/results`. Versionierte Python-Skripte leiten daraus Tabellen und Diagramme ab. Ursprüngliche und neue Datensätze werden niemals stillschweigend zusammengeführt. Die bereitgestellte ursprüngliche REIST-PDF und optionale FT2-Aufnahmen Dritter werden nicht weitergegeben.

`Build-Study.ps1` ist der dokumentierte Einstiegspunkt für Builds, neue Messverzeichnisse, Auswertung und LaTeX. Die eigenständigen Wellenformexperimente benötigen GNU C++, Boost-Header, CMake, Ninja und Python; die historische Reproduktion zusätzlich Fortran. Vollständige Decoder- und Profiltests benötigen den gesamten Fork und dessen reguläre Abhängigkeiten. Der Quellcode behält GPLv3 und die Urheberhinweise der Ausgangssoftware bei. Das Manuskript ist ein Entwurf; seine spätere Publikationslizenz ist noch nicht festgelegt. Im Rahmen dieser Studie

erfolgen weder neues Release noch DOI-Registrierung, externer Upload, Commit oder Push. Ein versioniertes Archiv und eine verifizierte DOI können später mit ausdrücklicher Zustimmung des Autors vorbereitet werden.

Autorenbeiträge und KI-Unterstützung

Rudolf Stepan stellte den REIST-Ansatz, den bestehenden Quellcode-Fork, die Forschungsziele und das Originalmanuskript bereit. OpenAI Codex unterstützte Quelltextprüfung, experimentelle Implementierung, lokale Ausführung, Auswertung, Diagramme und wesentliche Teile des Manuskriptentwurfs. Menschliche Verifikation und eine vom Autor verfasste Überarbeitung bleiben erforderlich. Diese Offenlegung ersetzt nicht die Einhaltung journalspezifischer Richtlinien. Die Arbeit würdigt die ursprünglichen Entwickler der WSJT-X-Protokolle und Software sowie die im Fork dokumentierten FT2-Beitragenden; deren Urheber- und Lizenzanerkennung wird nicht auf die Phasenzustandsanwendung übertragen.

Literatur

- [1] R. Stepan, *REIST-Division: Eine implementierungsorientierte Formulierung zentrierter Restarithmetik. Theorie, Architekturanalyse und empirische Evaluation auf ARMv8-A, x86-64 und FPGA*, bereitgestelltes Manuskript, 2026. DOI laut Manuskript: 10.5281/zenodo.21206471; öffentliche Metadaten nicht unabhängig abgerufen. SHA256 der Original-PDF im Audit.
- [2] R. Stepan, *REIST Division als Zustandsprinzip in den GFSK Rechenkernen von FT8 FT4 und FT2*, technischer Bericht v1.0, 15. Oktober 2026, docs/papers/reist-ftx/paper.tex, mit Commit 46ebbd78c archiviert.
- [3] Analog Devices, *MT-085: Fundamentals of Direct Digital Synthesis (DDS)*, technisches Tutorial. <https://www.analog.com/media/en/training-seminars/tutorials/mt-085.pdf>.
- [4] D. Goldberg, “What Every Computer Scientist Should Know About Floating-Point Arithmetic,” *ACM Computing Surveys*, Bd.23, Nr.1, S.5–48, 1991. Maßgeblicher Nachdruck: https://docs.oracle.com/cd/E19957-01/806-3568/ncg_goldberg.html.
- [5] G. E. Blelloch, *Prefix Sums and Their Applications*, Carnegie Mellon University, CMU-CS-90-190, November 1990. <https://www.cs.cmu.edu/afs/cs.cmu.edu/project/scandal/public/papers/CMU-CS-90-190.html>.
- [6] S. Franke, B. Somerville und J. Taylor, “The FT4 and FT8 Communication Protocols,” *QEX*, Juli/August 2020, S.7–17. https://wsjt.sourceforge.io/FT4_FT8_QEX.pdf.
- [7] Z. Cui, Z. Wang und X. Zhang, *Reduced-Complexity Column-Layered Decoding and Implementation for LDPC Codes*, arXiv:1204.2577, 2012. <https://arxiv.org/abs/1204.2577>.
- [8] Intel, *Intel 64 and IA-32 Architectures Software Developer’s Manual*, CPUID-Blatt 1AH, Aufzählung hybrider Kerntypen. <https://cdrdv2-public.intel.com/868137/325462-089-sdm-vol-1-1-2abcd-3abcd-4.pdf>.
- [9] Microsoft, *QueryThreadCycleTime function*, Win32-API-Dokumentation. <https://learn.microsoft.com/en-us/windows/win32/api/realtimeapiset/nf-realtimeapiset-querythreadcycletime>.