

HB radio

Swiss Radio Amateurs



**HB9BA/p - S. 35
Meisterleistung !**

**FACB - S. 50
10 Jahre Kurswesen**

**HB88YL - S. 56
Exklusiv nur YLs !**

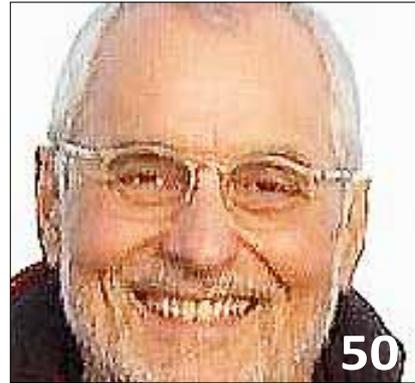
HB9BIN: 3000 Band- punkte übertroffen !



Jürg HB9BIN



Ronald OE3REB



Franz HB9EDU

Impressum

Organ der Union Schweizerischer Kurzwellen-Amateure
 Organe de l'Union des Amateurs Suisses d'Ondes courtes
 Organo dell'Unione Radioamatori di Onde Corte Svizzeri
 83. Jahrgang des *HBradio* [old man]
 83^e année de l' *HBradio* [old man]
 83. annata dell' *HBrado* [old man]
ISSN: 1662-369X

Auflage: 3'400 Exemplare
Herausgeber: USKA, 8804 Au
Geschäftsstelle: Willy Rüschi, HB9AHL, Bahnhofstrasse 26, 5000 Aarau, Tel: 079 842 65 59, E-Mail: gs@uska.ch
QSL-Service: Ruedi Dobler, HB9CQL, PF 816, 4132 Muttenz, Tel: 061 463 00 22
Redaktion/Layout: Willy Rüschi, HB9AHL, E-Mail: redaktion@uska.ch
Rédaction francophone: Werner Tobler, HB9AKN, Chemin de Palud 4, 1800 Vevey VD; Tel: 021 921 94 14; E-Mail: hb9akn@uska.ch
Webredaktor www.uska.ch: Josef Rohner, HB9CIC, E-Mail: webmaster@uska.ch

Eingesandte Texte können redaktionell bearbeitet werden. Bei grösseren Änderungen nehmen die Redaktionen Rücksprache mit den Autoren. Die einzelnen Artikel geben die persönliche Meinung der Autoren wieder. Redaktionen und USKA-Vorstand übernehmen dafür keine Verantwortung; es sei denn, dass ein Artikel ausdrücklich als offizielle Haltung der USKA bezeichnet wird.

Inserate und Hambörse: Yvonne Unternährer, HB9ENY, Dornacherstrasse 6, 6003 Luzern; Tel: 032 511 05 52; E-Mail: inserate@uska.ch

Bibliothek und Archiv: Philippe Schaetti, HB9ECP, Leimenweg 11, 4124 Schönenbuch, Tel: 061 302 14 00; E-Mail: biblio@uska.ch

Druck: Tisk Horák AG, Drážd'anská 83A, CZ - 400 07 Ústí nad Labem

Versand: Beorda AG, Kantonsstrasse 101, 6234 Triengen LU; E-Mail: mail@beorda.ch

Union Schweizerischer Kurzwellen-Amateure
 Union des Amateurs Suisses d'Ondes courtes
 Unione Radioamatori di Onde Corte Svizzeri

PC-Konto: 30-10397-0
 UBS Bern: IBAN CH46 0023 5235 6576 6740 K
 SWIFT: UBSWCHZH80A

Adressänderungen: uskadb@uska.ch

Titelbild
 Der 3000er Sticker des DXCC-Challenge Diplom von Jürg Regli HB9BIN ist selbst schon eine schöne Plakette
[Foto: Willy HB9AHL]

Inhalt - Table des matières

Thema - Thème - Tema	
Erfolgreich DXen und Jagd nach Bandpunkten	2
DXer avec succès et la chasse des points/bandes [F]	10
Fare DX con successo e la caccia dei punti banda [I]	18
HF-Activity	
Weihnachts-Contest 2015	24
Rückblick auf HB9ØIARU	30
HF-Contest-Calendar: February 2016 - April 2016	32
USKA KW/OC-Contest Calendar 2016	33
Das Pile-Up Gedicht	33
DX - IOTA - SOTA	
7J4AAL and his big gun	34
V51V – Vereinsmitglied von HB9Z	34
Ehrungen	
Sektion Solothurn: USKA-VHF-Diplome auf 1.3, 2.4 und 10 GHz	35
VHF - UHF - SHF	
VHF-Telegraphy-Contest/Marconi Memorial Contest Nov. 2015	39
USKA UKW/OUC-Contest Calendar 2016	41
HB9TV-ATV – Relais (ARALD)	41
Satelliten	
Satelliten / OSCAR-News	42
Technik - Technique	
Hinter den Kulissen des „System Fusion“ von Yaesu (2. Teil)	44
Magnetische Antennen-Kopplung	48
Sections - Clubs	
10 Jahre Kurswesen des FACB für die HB9-Prüfung	50
Sektion Luzern: 2016 wiederum ein CW-Kurs	52
Historik	
Notlandung des Helikopters HB-XCP im Dschungel von Surinam	54
USKA	
Jahresbericht 2015 von HB9O	56
HB88YL nur für YLs im August 2016	56
Antennenkommission in neuen Händen	57
Die neuen KW-Champions 2015	57
430-Tonnen-Flaggschiff mit einem Touch von Alinghi	58
Internationales	
IARU Region 1 News N° 195	60
60m-Band: Zuteilung von 15 KHz durch die WRC 2015	60
Mutationen	61
Hambörse	62
Redaktionsschluss, Inserate	63

Wir gratulieren herzlich:
Willi Vollenweider HB9AMC ist neuer USKA-Präsident !

Hinter den Kulissen des „System Fusion“ von Yaesu (2. Teil)

Mathias Weykand HB9FRV (Leiter Betrieb HB9UF)

Über den neuen „System Fusion“-Standard von Yaesu wird zurzeit viel diskutiert. Im Fokus dieser Diskussionen stehen vorwiegend die Sprachqualität, die Möglichkeiten der Verlinkung, das automatische Anzeigen von Rufzeichen und Distanzen zwischen zwei Stationen und ähnliches. Vergleiche mit anderen Betriebsarten fallen denn auch je nach Gesprächspartner ausgesprochen wohlwollend oder aber vernichtend aus. Relativ wenig gesprochen wird hingegen über die Art und Weise der Datenübertragung. Dieser zweiteilige Artikel nimmt sich dieser Thematik an (Teil 1 erschien im HBradio 6/2015).

Einführung

Im ersten Teil meines Artikels [1] erörtere ich Aspekte der digitalen Betriebsart YSF („Yaesu System Fusion“) anhand meines experimentellen YSF-Decoders [2]. Genauer betrachtet wurden die Modulation mit Frequenzumtastung und die Rekonstruktion der Dibits aus dem demodulierten Signal durch Abtastung zum richtigen Zeitpunkt (Sampling) sowie die Abbildung des abgetasteten Wertes auf ein Dibit (Slicing). Der Artikel endet mit der Darstellung eines auf diese Art aus dem Datenstrom gewonnenen YSF-Paketes. An dieser Stelle setze ich mit diesem zweiten Teil des Artikels ein.

Je nach Typ ist in diesem Paket nur Text (Rufzeichen, GPS-Koordinaten des Senders, etc.), nur Sprache oder beides enthalten. YSF bietet auch die Möglichkeit, Bilder zu versenden (nämlich im „DW“-Modus, Data-Wide). Wie dies genau funktioniert hat Yaesu in der Spezifikation jedoch nicht beschrieben. Ich gehe deshalb an dieser Stelle nicht weiter darauf ein; der Empfang von Bildern wird in meinem Decoder zurzeit auch nicht unterstützt. Ich kann mir aber nicht vorstellen, dass die Bildübertragung besonders ausgeklügelt ist – eine

Implementierung durch die Analyse von übertragenen Beispielbildern wäre deshalb vermutlich relativ einfach zu realisieren.

Abb. 1 zeigt ein vereinfachtes Verarbeitungsschema des Decoders. Es ist die Aufgabe des Deframers, die YSF-Pakete je nach Typ in ihre Bestandteile (Text und Sprache) aufzuspalten. Dies ist übrigens einer der Gründe, weshalb bei YSF die Wiedergabe leicht verzögert ist: Das System muss warten bis das ganze Paket empfangen wurde und kann es erst im Anschluss weiterverarbeiten. Die Text- und Sprach-Bestandteile werden dann weiterverarbeitet, decodiert und angezeigt bzw. als Audio ausgegeben. Im Folgenden betrachte ich einige Techniken, welche beim Verarbeiten eingesetzt werden. Es handelt sich dabei einerseits um vorwärts gerichtete Fehlerkorrektur („forward error correction“, FEC), andererseits um weitere Methoden, um die Übertragung robust gegenüber Störungen zu machen.

Multi-Band Excitation (MBE)

Bei der digitalen Übertragung von Sprache wird das analoge Signal abgetastet und anschliessend komprimiert, übertragen, und im Empfänger dann dekomprimiert und abgespielt. Diesem Schema folgen heute alle gängigen Betriebsarten wie D-STAR, DMR und P25 – und YSF bildet da keine Ausnahme. Die Komprimierung ist notwendig, denn die bei schmalbandigen Funkkanälen übliche Datenrate reicht nicht aus, um die Sprache unkomprimiert zu übertragen. Die zur Verfügung stehende Datenrate ist entsprechend derart beschränkt, dass sogar Komprimierungs-Verfahren wie z.B. MP3 bei weitem nicht ausreichen. Es kommen deshalb Komprimierungs-Verfahren zum Einsatz, welche speziell auf die Komprimierung von Sprache ausgelegt sind. Die Effizienz

dieser Verfahren ist mir erst richtig klar geworden, als ich auf der Projektwebseite meines Decoders [2] ein in MP3 komprimiertes Audio-Beispiel hochladen wollte. Während die MP3-Datei etwa 70 kB gross war belegten die aus den YSF-Paketen extrahierten Sprachdaten nur knapp 3 kB. Alle oben genannten digitalen Betriebsarten verwenden zum Erreichen dieser bemerkenswerten Kompressionsraten die Technologie der Firma „Digital Voice Systems, Inc.“ (DVSI) in Form von Chips oder lizenzierter Firmware. Diese Technologie wird im Überbegriff „Multi-Band Excitation“ (MBE) zusammengefasst und von DVSI stetig weiterentwickelt. Die Spezifikationen zu MBE sind mit wenigen Ausnahmen nicht öffentlich zugänglich – verständlicherweise möchte DVSI der Konkurrenz nicht in die Hände spielen. Eine Ausnahme bilden aber die aus rechtlichen Gründen öffentlich zugänglichen Standards der MBE-Varianten, welche in P25 zur Anwendung kommen. Dort wird genau die MBE-Variante spezifiziert, die auch bei DMR und teilweise bei YSF (nämlich im „DN“-Modus) zum Einsatz gelangt. Das Dokument [3] kostet zurzeit 256 USD. Eine Gruppe von Programmierern, welche aufgrund rechtlicher Gründe unbekannt bleiben möchte, hat mit Hilfe dieser Spezifikationen einen Decoder namens „mbelib“ implementiert, den ich in meinem Projekt verwende.

Die Alternative zu einer Software-Lösung ist das Beschaffen eines Vocoder-Chips von DVSI. Hierzu gibt es zwei Möglichkeiten: Entweder man besorgt sich den Chip auf einer Evaluierungsplatine, wie z.B. PiDV oder ThumbDV der Firma NW Digital Radio [4]. Diese Produkte sind sehr einfach anzuwenden aber mit rund 100 USD zuzüglich Postversand und MWST verhältnismässig teuer. Entgegen anderslautender Berichte ist

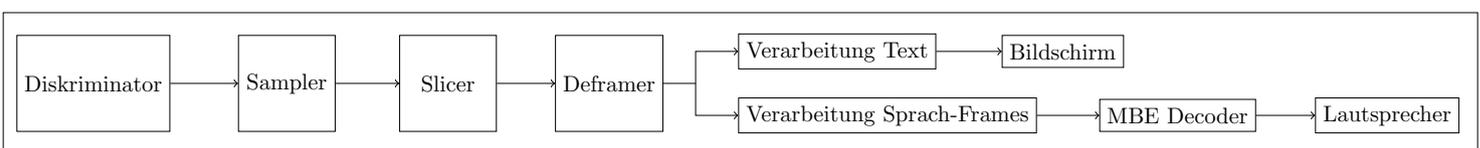


Abb. 1: Schema des YSF-Decoders

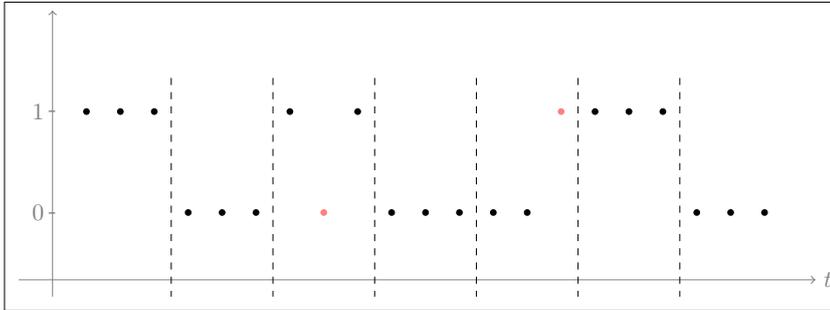


Abb. 2: Fehlerkorrektur durch Auswertung der Dreifach-Bits

es auch problemlos möglich den Chip als Privatperson in kleinen Mengen direkt bei DVSI einzukaufen. Ich habe das versucht und DVSI verkaufte mir die Minimalanzahl von 6 Chips zu je etwa 24 USD. Das ging erstaunlich reibungslos; man merkt aber, dass diese Firma auf das Beliefern von grossen Herstellern ausgelegt ist. Für die Überweisung des Betrages wurden mir in der Schweiz 20 CHF und auf der amerikanischen Seite nochmals 25 USD Bankspesen verrechnet und für die Lieferung in die Schweiz noch zusätzlich 115 USD Kosten in Aussicht gestellt. Ich habe deshalb die Chips an einen Freund in den USA liefern lassen und so einen Grossteil der Lieferkosten eingespart.

Diese Ausführungen zeigen, dass die Beschaffung der notwendigen Bauteile zwar möglich, ein Experimentieren mit dieser Technologie im Sinne unseres Hobbys jedoch aus meiner Sicht nicht realistisch ist. Wenn jemand z.B. einen Scanner für D-STAR, DMR oder Fusion (oder gar für alle zusammen!) bauen möchte, kann er sein Projekt nur verwirklichen solange DVSI nicht nur exklusiv grosse Hersteller beliefert. Er erhält in diesem Fall eine Art „Blackbox“ ohne Möglichkeiten Änderungen vorzunehmen. Für eine Auseinandersetzung mit der Technologie an sich sind daher vermutlich quell-offene und von Funkamateuren geleitete Projekte und Standards wie Codec2 [5] wesentlich geeigneter.

Forward Error Correction (FEC)

In einer analogen FM-Übertragung nimmt das Signal-Rausch-Verhältnis mit abnehmender Feldstärke ab – das Signal hört sich mehr und mehr verrauscht an, bis es unkenntlich im Rauschen untergeht. Bei digitalen Sprachmodi ist dies nicht der Fall: Die Qualität des Signals aus dem Lautsprecher bleibt ziemlich konstant bis es dann sehr abrupt absackt. Grund dafür ist, dass eine

Fehlerkorrektur zum Einsatz kommt; die Sprachdaten werden nämlich redundant verschickt. Fällt ein Teil dem Rauschen zum Opfer kann es unter Umständen aus noch intakten, redundanten Teilen rekonstruiert werden. In YSF kommen hierfür unter anderem ein Golay-Code und ein Faltungscode zur Anwendung. Es handelt sich hierbei um verbreitete Techniken, die in einführender Literatur zum Thema [6] umfassend diskutiert werden. Bei der Übertragung der MBE-Sprachdaten hat Yaesu einen interessanten Ansatz gewählt. Das Funkgerät lässt sich in einem VW- (Voice-Wide) und einem DN- (Data-Narrow) Modus betreiben. Im VW-Modus werden sehr vereinzelt Text (Rufzeichen etc.) und vorwiegend Sprache übertragen. Diese MBE-Sprachframes verfügen bereits über eine eingebaute Fehlerkorrektur. Im DN-Modus sind Daten und Sprache ineinander verwoben. Das in Teil 1 des Artikels abgedruckte Paket ist ein Beispiel hierfür. Bei diesen MBE-Sprachframes hat sich Yaesu entschieden anstelle der FEC von DVSI eine eigene FEC einzusetzen. Diese ist im Gegensatz zu den oben erwähnten Codes sehr einfach gehalten und in **Abb. 2** dargestellt: Jedes Bit wird dreimal wiederholt. Damit werden zwar dreimal so viele Daten übertragen, aber wenn ein Bit falsch übertragen wird kann es aufgrund der anderen beiden noch intakten Bits erkannt und korrigiert werden. Das funktioniert natürlich nicht mehr, wenn zwei oder gar alle drei Bits fehlerhaft sind.

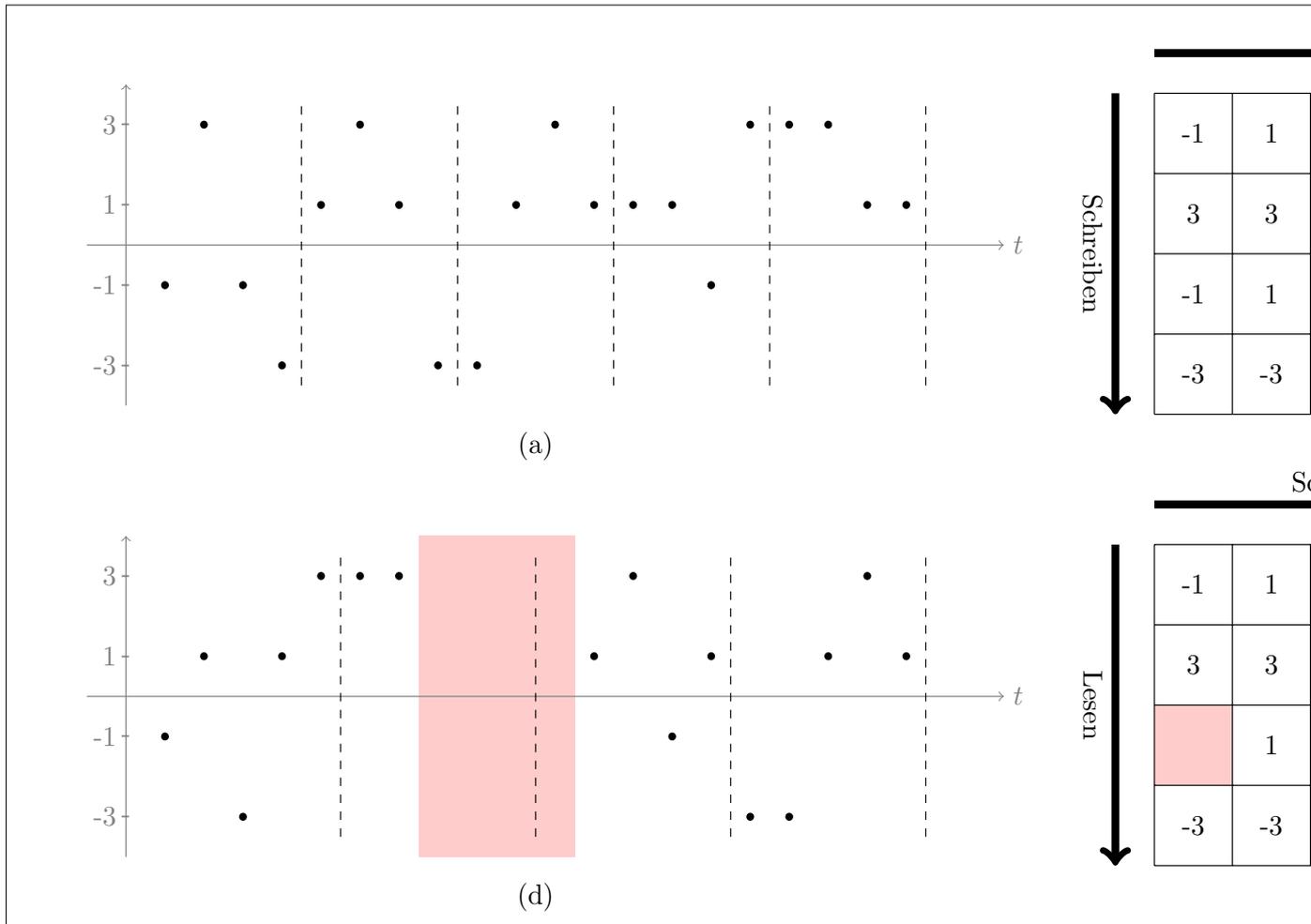
Das MBE-Sprachframe hat eine Grösse von 49 Bits. Das ganze Frame zu verdreifachen würde folglich 147 Bits belegen; es stehen aber aufgrund der beschränkten Bandbreite nur 104 Bits zur Verfügung. Deshalb haben sich die Ingenieure bei Yaesu dazu entschieden nur die ersten 27 Bits des MBE-Sprachframes zu verdreifachen (81 Bits); die restlichen 22 Bits werden belassen. Wer nach-

rechnet wird sehen, dass diese Vorgehensweise insgesamt 103 Bits belegt. Das letzte Bit wird deshalb stets mit einer ‚0‘ gefüllt. DVSI legt ihre Frames so aus, dass die ersten Bits die Wichtigsten und die Letzten die Unwichtigsten sind. Das Verdreifachen der ersten Bits macht in diesem Licht mehr Sinn. Unter „wichtig“ ist der Einfluss der Falschübertragung eines einzelnen Bits auf die Qualität zu verstehen. Ein Vergleich der Fehlerkorrektur durch Dreifach-Bits mit etablierten Verfahren (z.B. des von DVSI verwendeten Golay-Codes [6]) wäre ein interessantes Experiment.

Robustheit durch Interleaving

Die Methode der Verdreifachung ist gegenüber Störungen nicht robust. Störungen haben oft die Eigenschaft eine ganze Reihe von aufeinander folgenden Bits zu verfälschen. Somit schlägt die Fehlerkorrektur nach **Abb. 2** fehl. Dies gilt für andere FEC-Verfahren ebenso. Abhilfe schafft auf einfache Art und Weise das Abändern der Reihenfolge. In YSF geschieht dies auf der Ebene der Dibits, also noch vor der Aufspaltung der Dibits in einzelne Bits (siehe Teil 1 dieses Artikels [1]). Das folgende Beispiel ist in **Abb. 3** dargestellt: Es soll die Sequenz (-1, 3, -1, -3, 1, 3, 1, -3, -3, 1, ...) übertragen werden (**Abb. 3a**). Die Sequenz wird spaltenweise in eine Matrix mit vier Zeilen à fünf Spalten eingetragen (**Abb. 3b**). Die erste Spalte ist folglich von oben nach unten (-1, 3, -1, -3), die zweite Spalte ist (1, 3, 1, -3) usw. Die Spalten-Umbrüche sind in **Abb. 3a** mit gestrichelten Linien markiert. Ist die ganze Matrix voll wird sie zeilenweise ausgelesen und die so gelesenen Dibits übertragen. Diese übertragene Sequenz (-1, 1, -3, 1, 3, 3, ...) ist in **Abb. 3c** gezeigt – die gestrichelten Linien markieren nun die Zeilen-Umbrüche im Lesevorgang. In **Abb. 3d** ist die Sequenz am Empfänger zu sehen. Es handelt sich dabei um die übertragene Sequenz (**Abb. 3c**), wobei vier Dibits infolge einer Störung inkorrekt übertragen und rot markiert sind. Wie eingangs erwähnt würde so die Effektivität des FEC-Verfahrens empfindlich geschmälert. Aber im Empfänger wird die empfangene Dibit-Sequenz erneut durch die gleiche Matrix in die ursprüngliche

Hinter den Kulissen des „System Fusion“ von Yaesu (II)



Reihenfolge umgesetzt. Dazu wird nun zeilenweise geschrieben und spaltenweise gelesen (Abb. 3e; mathematisch entspricht dieser Vorgang der doppelten Transposition der Matrix). Die gestörten Dibits, in Abbildung 3f rot markiert, sind auf diese Weise so weit voneinander entfernt, dass sie als unabhängige Einzelfehler behandelt werden können.

Abb. 3: Zerstreuung eines Burst-Fehlers (rot) durch Verändern der Reihenfolge bei der Übertragung einer Dibit-Sequenz:

- (a): Zu übertragende Sequenz
- (b): Interleaving
- (c): Tatsächlich gesendete Sequenz
- (d): Empfangene Sequenz
- (e): De-Interleaving
- (f): Empfangene Sequenz nach Wiederherstellung der korrekten Dibit-Reihenfolge

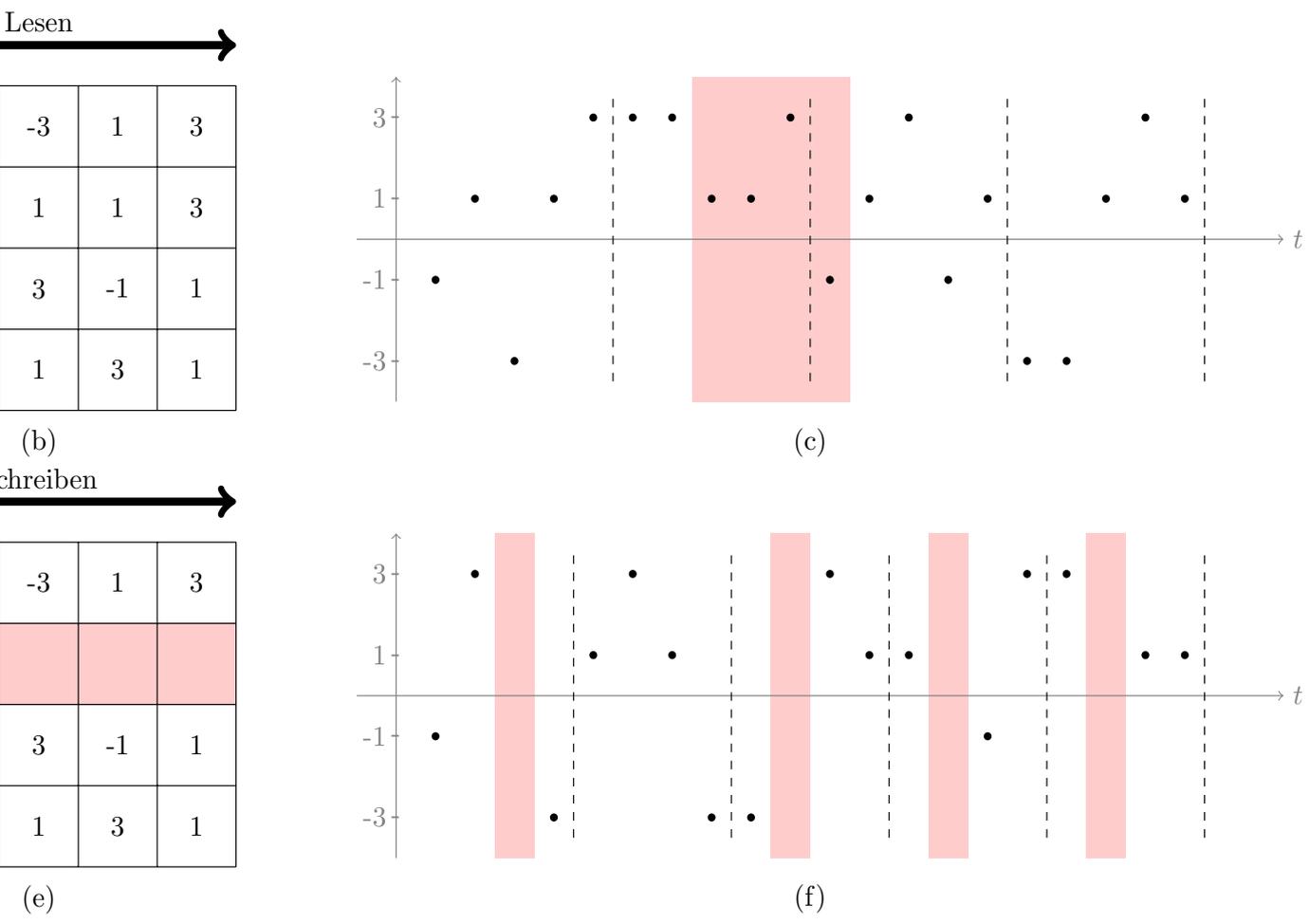
Ein wichtiges Detail möchte ich an dieser Stelle noch erwähnen: Im Abschnitt „Forward Error Correction (FEC)“ habe ich erwähnt, dass MBE-Sprachframes bereits Redundanz im Sinne einer FEC enthalten, diese aber im „DN“-Modus entfernt wird. Ich habe ebenfalls erwähnt, dass die ersten Bits des Frames wichtiger sind als die letzten Bits. Wer einen Chip von DVSI hat kann wählen wieviel Redundanz im MBE-Sprachframe eingefügt werden soll. Mehr Redundanz ist zwar robuster benötigt aber auch mehr Bandbreite. Es ist daher naheliegend für das Decodieren von MBE-Sprachframes im „DN“-Modus die Redundanz wegzulassen (auf 0 zu stellen), da diese ja stattdessen durch die Methode der Dreifach-Bits realisiert wird. Wird der Chip derart betrieben produziert er aber Frames, die mit YSF nicht kompatibel sind. Der Grund dafür ist, dass der Chip selbst ebenfalls Interleaving implementiert. Das ist natürlich nicht im

Sinne von Yaesu, denn Interleaving zerstört die Frame-Struktur insofern, als dass die ersten Bits nicht mehr die Wichtigsten sind. Bevor der Chip die MBE-Frames von YSF im „DN“-Modus verarbeiten kann müssen sie also erst in die richtige Reihenfolge gebracht werden. Dabei gibt es etwa $49! = 10^{62}$ mögliche Reihenfolgen (eine astronomisch grosse Zahl), aber nur eine davon ist die Richtige! Durch geschicktes Probieren liess sich die korrekte Reihenfolge aber relativ einfach eruieren; sie ist in Issue #12 meines Projektes [2] dokumentiert. Berichte über einen von Yaesu in diesem Mechanismus installierten kryptographischen Schlüssel kann ich weder nachvollziehen noch bestätigen.

Fazit

Ziel dieses Artikels ist es einzelne technische Aspekte der digitalen Betriebsart YSF zu erläutern. Ich verzichte in diesem Artikel absichtlich

Abbildung 3



auf eine Wertung bzw. einen Vergleich mit anderen digitalen Sprachübertragungsarten im Amateurfunk und habe mich ausschliesslich durch meine technische Neugierde leiten lassen. Fest steht, dass bei der Auseinandersetzung mit all diesen alternativen Betriebsarten zusätzlich zu elektronischen Fachkenntnissen über den Bau von Funkanlagen auch gewisse Grundkenntnisse in digitaler Signalverarbeitung notwendig sind. Ich hoffe, dass es mir gelungen ist aufzuzeigen, dass hier aber auch nur mit Wasser gekocht wird. Wie bereits mehrfach erwähnt ist mein Decoder ein Experiment und deshalb unvollständig. Mögliche Weiterentwicklungen wären der Empfang von Bildern oder das Auswerten von GPS-Koordinaten. Im Sinne des Open-source-Gedankens bin ich gerne bereit mit Interessierten zusammen an solchen Features zu arbeiten.

Dank

Beim Durchführen meines kleinen Experimentes wurde ich von vielen Personen tatkräftig unterstützt. Zuerst möchte ich den Vorstand von HB9UF nennen und insbesondere Peter Amsler HB9DWW und Hansjörg Baur HB9DWS hervorheben, die mir Fusion-Funkgeräte ausgeliehen und diesen Artikel korrekturgelesen haben. David Rickett VA7EEX hat mir freundlicherweise Zugang zu seinem DVSI-Chip gegeben damit ich einige Theorien überprüfen konnte. Johnathan Corgan, ein Kernentwickler von GNURadio, stand mir bei meinen ersten Schritten mit GNURadio zur Seite. Seine Analyse der C4FM-Modulation in DMR und P25 war ein guter Start in dieses Projekt. Sylvain Munaut hat mit mir hilfreiche Details in Bezug auf die Struktur von MBE-Sprachframes geteilt.

Bibliographie

- [1] *Hinter den Kulissen des „System Fusion“ von Yaesu (1. Teil)* Mathias Weyland HB9FRV, HBradio 6/2015
- [2] *gr-ysf Projektwebseite:* <https://hb9uf.github.io/gr-ysf>
- [3] *APCO Project 25 Half-Rate Vocoder Addendum – TIA-102.BABA-1*, APIC Vocoder Task Group, Telecommunications Industry Association.
- [4] *Produkte von NW Digital Radio,* <http://nwdigitalradio.com/products>
- [5] *Codec2 Projektwebseite von David Rowe, VK5DGR:* www.rowetel.com/blog/?page_id=452
- [6] *Digital Communications,* John G. Proakis & Masoud Salehi, int. ed. McGraw-Hill, 2008