



(19)
Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) DE 603 07 800 T2 2007.10.11

(12)

Übersetzung der europäischen Patentschrift

(97) EP 1 487 118 B1

(21) Deutsches Aktenzeichen: 603 07 800.1

(96) Europäisches Aktenzeichen: 04 018 527.4

(96) Europäischer Anmeldetag: 19.03.2003

(97) Erstveröffentlichung durch das EPA: 15.12.2004

(97) Veröffentlichungstag

der Patenterteilung beim EPA: 23.08.2006

(47) Veröffentlichungstag im Patentblatt: 11.10.2007

(51) Int Cl.⁸: H03M 13/39 (2006.01)
H04L 1/00 (2006.01)

(30) Unionspriorität:

115967 05.04.2002 US

(84) Benannte Vertragsstaaten:
DE, FR, GB

(73) Patentinhaber:

Lucent Technologies Inc., Murray Hill, N.J., US

(72) Erfinder:

Das, Arnab, Hudson, New Jersey 07302, US; Khan,
Farooq Ullah, Manalapan New Jersey 07726, US;
Sampath, Ashwin, Somerset New Jersey 08873,
US; Su, Hsuan-Jung, Monmouth NJ 07747, US

(74) Vertreter:

derzeit kein Vertreter bestellt

(54) Bezeichnung: Fehlererkennungsverfahren in drahtlosen Kommunikationssystemen

Anmerkung: Innerhalb von neun Monaten nach der Bekanntmachung des Hinweises auf die Erteilung des europäischen Patents kann jedermann beim Europäischen Patentamt gegen das erteilte europäische Patent Einspruch einlegen. Der Einspruch ist schriftlich einzureichen und zu begründen. Er gilt erst als eingelebt, wenn die Einspruchsgebühr entrichtet worden ist (Art. 99 (1) Europäisches Patentübereinkommen).

Die Übersetzung ist gemäß Artikel II § 3 Abs. 1 IntPatÜG 1991 vom Patentinhaber eingereicht worden. Sie wurde vom Deutschen Patent- und Markenamt inhaltlich nicht geprüft.

Beschreibung**HINTERGRUND DER ERFINDUNG****1. Technisches Gebiet**

[0001] Die Erfindung betrifft allgemein drahtlose Kommunikationssysteme und insbesondere Fehlerdetektionsverfahren zur Auswertung von in Steuerkanälen solcher Systeme übertragenen Informationen.

2. Allgemeiner Stand der Technik

[0002] Fast alle drahtlosen Kommunikationssysteme verwenden Kommunikation auf „Rahmenbasis“, wobei eine bestimmte Anzahl von Bit, die als ein Rahmen definiert ist, zusammen kanalcodiert und übertragen wird. Die meisten Systeme verwenden verkettete Codierung für jeden Rahmen mit einem inneren Fehlerkorrekturcode, wie zum Beispiel einem Faltungs- oder Turbo-Fehlerkorrekturcode, und einem äußeren Fehlerdetektionscode.

[0003] [Fig. 1](#) zeigt einen Prozeßfluß zur Bildung einer typischen verketteten Codestruktur in dem Sender einer Basisstation. Wie gezeigt wird bei **110** ein Fehlerdetektionscode zu einem Rahmen eines Datenkanals hinzugefügt, wobei der Rahmen k Bit umfaßt. In der Regel wird als der Fehlerdetektionscode ein CRC-Code (Cyclic Redundancy Check) verwendet, der hier mit einer Länge von p gezeigt ist. Die CRC-Bit werden auf der Basis der k Informationsbit berechnet. Der CRC-Code wird an den Rahmen angehängt (z.B. k + p Bit) und dann bei **120** durch Fehlerkorrekturcodierung geleitet. Der Fehlerkorrekturcode ist zum Beispiel ein Faltungscode mit einer Rate von 1/r, mit r > 1. Nach Fehlerkorrekturcodierung ist die Anzahl der Bit gleich (r·(k + p)), und sie werden dann zu einem Modulator geleitet und über einen Kanal übertragen. Ein Fehlerkorrekturcode-Decodierer im Empfänger eines Benutzergeräts (z.B. einer Mobilstation) versucht dann, etwaige auf dem Kanal stattfindende Bitfehler zu korrigieren.

[0004] In den meisten Fällen wird ein Rahmen verworfen, wenn der Empfänger in der Übertragung auf der Basis des Fehlerdetektionscodes einen Fehler erkennt, der unkorrigierbar ist. Dies führt zu einem Verlust oder einer Verzögerung von Informationen, abhängig davon, ob anschließend eine Neuübertragung ausgeführt wird. Der am häufigsten verwendete Fehlerdetektionscode ist der obenerwähnte CRC-Code. Standard-CRC-Codes enthalten Bitlängen von 8, 12, 16, 24 und/oder 32 Bit. Die Leistungs- oder Gütezahl bei Fehlerdetektionscodes ist die Wahrscheinlichkeit eines unerkannten Fehlers, d. h. ein Fall, in dem die Verwendung des inneren Fehlerkorrekturcodes Übertragungsfehler nicht korrigieren konnte und der äußere Fehlerdetektionscode nicht erkannt hat, daß die decodierten Informationen feh-

lerhaft waren. Dies ist ein unerkannter Fehler, weil die decodierten Informationen fehlerhaft sind, aber der Fehlerdetektionscode den Fehler nicht entdeckt hat. Die unerkannte Fehler-Wahrscheinlichkeit bei CRC-Codes beträgt in der Regel in der Ordnung von 2^{-L} , wobei L die Länge des CRC ist. Ein 8-Bit-CRC weist somit eine unerkannte Fehler-Wahrscheinlichkeit von ungefähr 1/256 auf.

[0005] Das mit der Verwendung eines CRC assozierte Overhead hängt von der Anzahl der Informationsbit in dem Rahmen ab. In der Regel übersteigt die Anzahl der Informationsbit jedes Rahmens, wie etwa des Rahmens k in [Fig. 1](#), mehrere hundert Bit, und somit ist ein etwaiger Overheadeffekt der Verwendung eines 8-, 12- oder 16-Bit-CRC minimal. Bei bestimmten Anwendungen müssen jedoch nur sehr wenige Bit pro Rahmen übertragen werden und das Overhead durch die Verwendung auch nur eines CRC der Länge 8 kann zu groß sein. Ein solches Beispiel ist die Übertragung von Steuerkanalinformationen in drahtlosen schnellen Datenkommunikationssystemen, wie zum Beispiel bei der HSDPA-Spezifikation (High Speed Downlink Packet Access) des UMTS-Standards (Universal Mobile Telecommunication System).

[0006] Bei HSDPA werden Übertragungsdaten für mehrere Benutzergeräte (im folgenden UE, die häufig auch als Mobilstationen bekannt sind) auf einem gemeinsamen schnellen gemeinsam benutzten Abwärtsstreckendatenkanal (HS-DSCH) gemultiplext. Man erhält hohe Datenraten durch Ablaufplanung, adaptive Modulation und Codierung und hybride automatische Wiederholungsanforderung (HARQ), wie bekannt ist. UE werden auf dem gemeinsam benutzten Datenkanal eingeteilt. Die UE werden entweder auf rein zeitlich gemultiplexte Weise (TDM), wobei alle verfügbaren Betriebsmittel (Leistungsbetriebsmittel und Datenkanalisierungscodes) während eines Übertragungszeitintervalls einem UE zugewiesen werden, eingeteilt, oder zwischen mehreren UE in einem Übertragungszeitintervall (TTI). Bei der Übertragung zu mehreren UE in einem TTI werden die Leistungsbetriebsmittel und Datenkanalisierungscodes unter diesen UE nicht unbedingt auf gleichförmige Weise aufgeteilt. Bei dem UMTS-Standard beträgt das Übertragungszeitintervall (TTI) in der Regel 2 ms oder 3 Zeitschlüsse (wobei jeder Zeitschlitz ungefähr 0,667 ms beträgt). Die Einteilung für die UE wird in der Regel auf der Basis eines bestimmten Typs von Informationen über die von dem UE erfahrene Kanalqualität erreicht.

[0007] Eine wichtige Komponente dieser schnellen drahtlosen Systeme ist die Verwendung eines Steuerkanals. Der Steuerkanal führt Informationen in bezug auf (a) welche UE zum Empfangen einer Datenübertragung über einen entsprechenden HS-DSCH eingeteilt wurden, (b) welche Datenkanalcodes je-

dem bestimmten UE zugewiesen werden und (c) modulations- und HARQ-bezogene Informationen. Vom Standpunkt der Systemeffizienz aus gesehen werden einige wenige Steuerkanäle so definiert, daß sie gemeinsam zwischen allen UE benutzt werden, anstatt einen eigenen Steuerkanal pro UE bereitzustellen.

[0008] Eine beispielhafte Konfiguration besteht darin, bis zu M schnelle gemeinsam benutzte Steuerkanäle (HS-SCCH) für gleichzeitige Übertragungen zu definieren, wobei zum Beispiel $M = 4$ ist. Für jedes TTI führt jeder HS-SCCH HS-DSCH-bezogene Abwärtsstreckenzeichenangabe für ein UE. Die Anzahl der HS-SCCH kann von einem Minimum von einem HS-SCCH ($M = 1$) bis zu einem Maximum von vier HS-SCCH ($M = 4$) reichen. Dies ist die vom Standpunkt des UE aus gesehene Anzahl der HS-SCCH. Anders ausgedrückt, bestimmt ein UE, ob eine nachfolgende Übertragung auf irgendeinem der HS-DSCH für es bestimmt ist oder nicht erst bei oder nach der Decodierung von Informationen in den HS-SCCH.

[0009] [Fig. 2](#) zeigt die Beziehung zwischen HS-SCCH **210** und ihren entsprechenden gemeinsam benutzten HS-DSCH-Gegenstücken **220**. In [Fig. 2](#) führt jeder HS-SCCHx ($x = 1$ bis 4) Informationen betreffend eines entsprechenden HS-DSCHx ($x = 1$ bis 4). Die Anzahl der HS-DSCH und deshalb die Anzahl der HS-SCCH, die verwendet werden kann, kann für jedes TTI abhängig von der Anzahl der gleichzeitig in dem TTI eingeteilten UE variieren. Die Konfiguration von HS-SCCH und HS-DSCH in [Fig. 2](#) ermöglicht folglich die Aufteilung der Datenkanalisierungscodes und Leistungsbetriebsmittel auf vier gleichzeitige Übertragungen.

[0010] Wieder mit Bezug auf [Fig. 2](#) werden Steuerkanaldaten auf jedem HS-SCCH in der Regel in zwei Teile unterteilt. Teil I besteht aus Informationen bezüglich der Datenkanalisierungscodes, die zum Beispiel einem bestimmten UE zugewiesen wurden. Daten von Teil II enthalten HARQ-bezogene Informationen und andere Transportinformationen. Um die Komplexität im UE niedrig zu halten, erlauben HS-SCCH-Entwürfe in der Regel ein Übertragen von Informationen von Teil I vor dem Beginn (d. h. vor $t = 0$) der Datenübertragung, wie in [Fig. 2](#) gezeigt. Bei der derzeitigen Konfiguration muß folglich jedes UE Teil I auf jedem HS-SCCH und in jedem TTI decodieren, um zu bestimmen, (a) ob die Übertragung für dieses konkrete UE bestimmt war oder nicht und (b) wenn die Übertragung für dieses bestimmte UE bestimmt war, muß das UE Teil I decodieren und herausfinden, auf welchen Kanalisierungscodes der entsprechende HS-DSCH ankommen wird.

[0011] Deshalb muß jedes UE in jedem TTI vor dem Beginn der Datenübertragung bis zu (4) HS-SCCH

decodieren. Vom Standpunkt der UE-Bearbeitungskomplexität aus gesehen ist es deshalb wünschenswert, die Anzahl der Bit in Teil I, die Bearbeitung erfordern, zu begrenzen, und es ist außerdem wünschenswert, daß die Verarbeitung so einfach wie möglich ist. Gleichzeitig sollten zwei Bedingungen für jedes UE erfüllt werden. Die erste besteht darin, daß in jedem UE die Wahrscheinlichkeit der Fehlerdetektion hoch sein sollte. Anders ausgedrückt, decodiert, wenn eine Übertragung für ein bestimmtes UE bestimmt ist, dieses UE Teil I und erkennt erfolgreich, daß die nachfolgende Datenübertragung auf dem entsprechenden HS-DSCH für diese UE bestimmt ist. Die zweite Bedingung, die zu erfüllen ist, besteht darin, daß eine Wahrscheinlichkeit eines Falschalarms niedrig ist. Ein Falschalarm liegt vor, wenn ein UE Teil I decodiert und falscherweise erkennt, daß die nachfolgende Datenübertragung auf einem entsprechenden HS-DSCH für dieses UE bestimmt ist.

[0012] Eine niedrige Wahrscheinlichkeit der Detektion hat verschwendete Betriebsmittel zur Folge, da jedes verpaßte Detektionsereignis bedeutet, daß die Übertragung auf dem entsprechenden HS-DSCH vergeudet ist. Ein Falschalarmereignis würde bewirken, daß ein UE, das NICHT für eine bestimmte Übertragung eingeteilt ist, damit beginnt, Daten zu puffern und zu versuchen, die Informationen zu decodieren, wodurch in dem UE zum Beispiel aufgrund unnötiger Verarbeitung Batteriebetriebsmittel verschwendet würden.

[0013] Um sicherzustellen, daß die beiden obigen Bedingungen erfüllt werden, wird in der Regel ein UE-spezifischer CRC-Code für die Fehlerdetektion am Teil I verwendet. Folglich decodiert das UE Bit von Teil I und wendet seine eindeutige CRC an, um nach Fehlern zu prüfen. Wenn Fehler vorliegen, nimmt das UE an, daß die Übertragung nicht für es bestimmt ist. Wenn die Verwendung des CRC keine Fehler am Teil I der HS-SCCH erkennt, decodiert das UE Teil II des HS-SCCH und beginnt mit dem Puffern und Decodieren des entsprechenden HS-DSCH.

[0014] Standard-CRC-Codes mit Bitlängen von 12 Bit oder mehr erzielen gewöhnlich eine akzeptable Detektions-/Falschalarmleistung. Teil I des HS-SCCH enthält jedoch gewöhnlich nur etwa 8 Informationsbit, die Verwendung eines 12-Bit-CRC oder mehr repräsentiert also ein wesentliches Overhead ($>150\%$). Darüber hinaus ist die Anzahl der Bit von Teil I, die das UE verarbeiten muß, zu groß. Zum Beispiel muß das UE für jeden HS-SCCH mit 8 Informationsbit und 12 CRC-Bit, um die Informationen von Teil I für vier HS-SCCH zu decodieren, 80 Datenbit in der Regel innerhalb eines Zeitschlitzes eines TTI (0,667 ms) verarbeiten. Dies ist unerwünscht und kommt im wesentlichen der Verarbeitung einer Spitzendatenrate von 120 kbps nur zur Decodierung eines Teils der Steuerinformationen gleich.

[0015] Eine herkömmliche Lösung besteht darin, die Verwendung eines CRC zur Fehlerdetektion zu vermeiden und statt dessen einen UE-spezifischen Verwürfelungs- oder Maskierungsansatz zu verwenden, der durch Berechnung einer spezifischen Faltungsdecodermetrik zur Fehlerdetektion befolgt wird, der nachfolgend ausführlicher beschrieben werden wird. Das Prinzip hinter diesem Ansatz besteht darin, daß, wenn das UE eine Übertragung, die für es selbst bestimmt ist, entwürfelt, die resultierende Decodermetrik gewöhnlich hoch ist. Immer dann, wenn das UE eine Übertragung entwürfelt, die für ein bestimmtes anderes UE bestimmt war, ist die Decodermetrik jedoch gewöhnlich niedrig. Das Verwürfeln/Entwürfeln zur Berechnung einer Decodermetrik, gefolgt durch einen Vergleich der Decodermetrik mit einer Schwelle ist deshalb ein Verfahren zur Erzielung einer Fehlerdetektion, wenn kein CRC verwendet wird.

[0016] [Fig. 3](#) zeigt einen Prozeßfluß, wie ein Verwürfelungscode mit einem Faltungsfehlerkorrekturcode verwendet wird. Der UE-spezifische Verwürfelungs-(Maskierungs-)Code „flippt“ bestimmte Bit (1 zu 0 und 0 zu 1) auf der Basis der dem UE zugewiesenen eindeutigen Verwürfelungssequenz.

[0017] Zum Beispiel nehme man in [Fig. 3](#) an, daß die Verwürfelungssequenz in Block **320** 0101 ist und die Ausgangsbit aus dem Faltungscode in Block **310** 1101 sind. Wenn der Verwürfelungscode die Ausgangsbit flippt, lautet die Sequenz nach Verwürfelung 1000. Wenn das UE diese Sequenz decodiert, invertiert das UE die Verwürfelung unter Verwendung seiner eigenen Sequenz und leitet die resultierenden Informationen dann durch einen Faltungsdecoder. Der Effekt dieser Verwürfelung besteht darin, daß, wenn die Übertragung nicht für das UE bestimmt ist, unter Verwendung eines Viterbi-Decodierungsalgorithmus des Faltungscodes berechnete Wegmetriken relativ niedrig sind. Es folgt eine Besprechung des Viterbi-Algorithmus, der von dem obenerwähnten UE-spezifischen Verwürfelungs- oder Maskierungsansatz zur Fehlerdetektion verwendet wird, sowie eine Besprechung von Wegmetriken.

Der Viterbi-Algorithmus

[0018] Der Viterbi-Algorithmus für die Faltungsdecodierung ist ein bekannter Decodierungsalgorithmus, der insofern optimal ist, als er die Maximalwahrscheinlichkeits(ML-) oder wahrscheinlichste Sequenz von Bit auf der Basis der Verwendung von Ausgangswerten aus dem Kanal ergibt. Viterbi-Decodierung ist die Standardtechnik zum Decodieren von Faltungscodes, gleichgültig, ob ein CRC verwendet wird oder nicht. Eine Beschreibung des Viterbi-Algorithmus zur Decodierung von Faltungscodes findet sich in einem Standardlehrbuch der Kommunikationstechnik, wie zum Beispiel „Digital Communications“ von J.G. Proakis, 2te Auflage, McGraw Hill. Ei-

nige der Konzepte werden hier nachfolgend der Zweckmäßigkeit halber kurz wiederholt.

[0019] Ein Faltungscodierer umfaßt eine Anzahl von Schieberegistern oder Speicherelementen. Die Anzahl der Schieberegister wird als die Nebenbedingungslänge des Codes bezeichnet und jedes Schieberegister speichert exakt 1 Informationsbit. Jedes Mal, wenn ein neues Bit hereinkommt, wird es in eine äußerste linke Schieberegisterspeicherstelle eingelesen, und der Inhalt jedes Schieberegisters wird in das Schieberegister unmittelbar rechts verschoben. Der Inhalt des äußersten rechten Schieberegisters wird offensichtlich weggeworfen. Ein Faltungscodierer kann also als an Bit operierendes Linearfilter angesehen werden.

[0020] Ein Faltungscodierer ist auch durch eine Coderate gekennzeichnet. Im allgemeinen definiert die Coderate, wie viele Ausgangsbit für jedes Eingangsbit produziert werden. Eine Coderate von $1/r$ bedeutet deshalb, daß für jedes eingegebene Informationsbit r codierte Bit durch den Codierer ausgegeben werden. Je größer der Wert von r ist, desto leistungsstärker ist also der Code (d. h. desto größer seine Fähigkeit zur Korrektur von Übertragungsfehlern). Wie die r Ausgangsbit produziert werden, muß schließlich spezifiziert werden; dies wird durch die Verbindungen von Schieberegisterelementen mit Exklusiv-OR-Elementen gegeben.

[0021] [Fig. 4](#) ist ein Zustandsdiagramm eines einfachen Faltungscodierers **400**. Der Faltungscodierer **400** besitzt eine Nebenbedingungslänge $k = 3$ und eine Rate $1/2$, wie bekannt. Die Nebenbedingungslänge definiert die Anzahl der verwendeten Schieberegisterelemente. Die Schieberegisterelemente sind Teil eines Speichers oder Schieberegisters in dem UE. In [Fig. 4](#) liegen drei Schieberegisterelemente **410**, **420** und **430** vor. Das Schieberegisterelement **410** enthält das zuletzt eingefügte Informationsbit. Jedes Mal, wenn ein neues Informationsbit hereinkommt, werden die vorherigen Bit nach rechts verschoben, so daß das mittlere Element **420** das zweitneueste Bit und das äußerste rechte Bit **430** das drittneueste Bit (z.B. das älteste Bit) enthält. XOR-Operationen bei **440** und **450** bestimmen, wie die codierten Bit bestimmt werden. Jedes Mal, wenn ein Informationsbit **415** eingegeben wird, werden wie in [Fig. 4](#) gezeigt zwei codierte Bit **445** und **455** ausgegeben.

[0022] [Fig. 5](#) zeigt ein Trellisdiagramm **500** zur Erläuterung, wie der Viterbi-Algorithmus und insbesondere die Viterbi-Decodierung in dem UE implementiert wird. In [Fig. 5](#) als schwarze Punkte repräsentierte Zustände in den Trellis bezeichnen die vier möglichen Schieberegisterinhalte (00, 01, 10 oder 11, entlang der linken Seite des Trellisdiagramms gezeigt), wenn ein Bit hereinkommt. Für binäre Faltungscodes beträgt die Anzahl der Zustände deshalb

$2^{\text{Nebenbedingungslänge}-1}$ oder 2^{k-1} . Für jedes eingegebene Informationsbit befänden sich das mittlere Schieberegisterelement **420** und das äußerste rechte Element **430** deshalb abhängig von dem Zustand der vorherigen beiden Informationsbit, die rechts geschoben wurden, in einem Zustand von 00, 01, 11 oder 10. Das erste Bit in dem Zustand ist als das älteste Bit (innerhalb des äußersten rechten Schieberegisters **430**) definiert, und das zweite Bit (Inhalt des mittleren Schieberegisters **420**) als das zweite nächstälteste Bit. Wenn also die zwei vorherigen Bit 1 und 0 waren, ist der Zustand dann 01.

[0023] Das Trellisdiagramm wird für die in [Fig. 4](#) gezeigte Faltungscodierung angegeben und ist für sieben (7) Ebenen (Ebenen werden durch J repräsentiert, mit $J = 1$ bis 7), entsprechend 7 Informationsbit. Das Trellisdiagramm in [Fig. 5](#) nimmt an, daß zwei (2) Nachspannbit verwendet werden, um den Trellis abzuschließen und den Decoder zu dem Zustand 00 zurückzuführen. Im allgemeinen wäre die erforderliche Anzahl von Nachspannbit gleich der Nebenbedingungslänge minus 1 ($k - 1$).

[0024] Jede Ebene J ($J = 1$ bis 7) über den oberen Teil des Trellisdiagramms in [Fig. 5](#) hinweg entspricht jedem Informationsbit. Vor dem Eingeben des ersten Bit werden die beiden Schieberegisterelemente **420** und **430** immer auf 0 gesetzt, so daß der Startzustand auf der Ebene $J = 0$ immer 00 ist. Wenn das erste Informationsbit eine 0 ist, bleibt auf der Ebene $J = 1$ der Decoder im Zustand 00. wenn dagegen das erste Informationsbit in dem Decoder **400** bei $J = 0$ eine 1 ist, bewegt sich der Decoder **400** auf der Ebene $J = 1$ zum Zustand 01. Jeder mögliche Übergang von einer Ebene zur nächsten wird als ein Zweig bezeichnet. Die Sequenz verbundener Zweige wird als ein Weg durch den Trellis bezeichnet.

[0025] Der Viterbi-Algorithmus schreitet voran, indem auf jeder J-Ebene eine sogenannte „Zweigmetrik“ berechnet wird. Man erhält die Zweigmetrik durch „Korrierieren“ der empfangenen Bit aus dem Kanal mit den Bit, die gesendet worden wären, wenn man diesen Zweig als den korrekten Kanal für das UE genommen hätte. Je höher die Korrelation, desto wahrscheinlicher war der Zweig der korrekte. In einem Beispiel, bei dem die tatsächlichen empfangenen Bit aus dem Kanal für eine bestimmte J-Ebene 10 sind, weist also dann jeder Zweig, der eine Zweigmetrik von 10 auf dieser J-Ebene produziert hätte, die höchste Korrelation auf.

[0026] Zweigmetriken aus vorherigen J-Ebenen werden summiert, um eine sogenannte Wegmetrik zu ergeben. Je höher der Wert der Wegmetrik ist, desto wahrscheinlicher ist es, daß die Wegmetrik der tatsächlichen übertragenen Bitsequenz entspricht. In jedem Zustand in dem Trellis und auf jeder Ebene werden zwei Wege zusammenlaufen. Viterbi-Decodie-

rung bringt deshalb mit sich, die Wegmetriken der beiden zusammenlaufenden oder konkurrierenden Wege zu vergleichen und den mit der schlechtesten (niedrigwertigsten) Wegmetrik zu verwerfen. Aufgrund von Rauschen wird gelegentlich während eines der obenerwähnten Zusammenlauffälle der falsche Weg statt dem richtigen gewählt. Dies führt zu einem Decodierungsfehler. Der durch den Viterbi-Algorithmus ausgewählte Weg wird folglich als der überlebende oder Gewinnerweg bezeichnet.

[0027] Es gibt pro Ebene höchstens einen Gewinnerweg pro Zustand. Da Nachspannbit verwendet werden, ist der Zustand auf der letzten Ebene immer ein Zustand mit nur Nullen. In dem Beispiel werden 2 Nachspannbit verwendet und der Zustand auf der letzten Ebene ist 00. Wie bereits erwähnt, laufen in dem Zustand 00 auf der letzten Ebene zwei Wege zusammen. Folglich wählt der Viterbi-Decodierungsalgorithmus den Weg mit der größeren Wegmetrik auf dieser letzten Ebene ($J = 7$). Dieser weg ist nun der letztendliche Gewinnerweg und repräsentiert die „wahrscheinlichste“ Sequenz übertragener Bit.

[0028] Bei dem Verwürfungsansatz, bei dem kein CRC zur Fehlerdetektion verwendet wird, wird Fehlerdetektion folglich unter Verwendung von Viterbi-Decodierung mit dem Verwürfungsansatz durchgeführt. In diesem Kontext besteht das herkömmliche Verfahren darin, eine Wegmetrikdifferenz zwischen den zusammenlaufenden Wegen auf der letzten Ebene (z.B. $J = 7$ in dem Trellisdiagramm von [Fig. 5](#)) zu bestimmen. Wenn diese berechnete Wegmetrikdifferenz unter einer vorbestimmten Schwelle liegt, wird die Decodierung als unzuverlässig betrachtet und das UE deklariert, daß die Übertragung auf diesem HS-SCCH nicht für es bestimmt war. Dies wird als Ansatz der Endwegmetrikdifferenz (EPMD) zur Fehlerdetektion bezeichnet. Der EPMD-Ansatz vermeidet die Übertragung von CRC-Bit und verringert somit die Verarbeitungsanforderung im UE.

[0029] „HS-SCCH: Performance Results and Improved Structure“ zeigt die Leistungsfähigkeit eines ersten Teils (Teil 1) einer herkömmlichen HS-SCCH-Decodierungsstruktur über die Wahrscheinlichkeit P_m und die Wahrscheinlichkeit eines Falschalarms P_f im Vergleich zu dem zweiten Teil (Teil 2) des HS-SCCH. Wie dargestellt, wird ein nachfaltungscodierter Teil 1 durch eine UE-ID verwürfelt, und die UE-ID-Basisverwürfungssequenz wird in zwei Stufen erzielt, indem zuerst die 10-Bit-UE-ID unter Verwendung eines aus einem Subcode zweiter Ordnung abgeleiteten Blockcodes zum Erhalten von 32 Bit codiert und dann die ersten 8 Bit des 32-Bit-Ergebnisses wiederholt werden, um die übrigen 8 Bit zu produzieren.

[0030] Motorola und Philips: „Way Forward on HS-SCCH“, TDOC R1-02-0463 von 3GPP TSG RAN WG 1, <<http://www.3gpp.org>>, 18–22.2.2002, Seiten

1–2, betrifft ein Codierungsschema für gesteuerte Kanalzeichengabe für einen HS-DSCH (schnellen gemeinsam benutzten Abwärtsstreckenkanal) in bezug auf die Decodierung des gemeinsam benutzten gesteuerten Kanals in dem UE. Das UE entwürfelt einen Teil jedes gemeinsam benutzten Steuerkanals, und eine Metrikdifferenz zwischen den wahrscheinlichen Wegen während des Viterbi-Decodierungsprozesses wird als zuverlässige Grundlage für die Validität eines gemeinsam benutzten Steuerkanals genommen.

Kurfassung der Erfindung

[0031] Ein Verfahren gemäß der Erfindung wird in Anspruch 1 definiert. Bevorzugte Formen werden in den abhängigen Ansprüchen definiert.

[0032] Es wird ein effizientes und zuverlässiges Verfahren zum Erkennen von Fehlern in Steuerkanälen eines drahtlosen Kommunikationssystems bereitgestellt, um sicherzustellen, daß für jedes UE in dem System die Wahrscheinlichkeit der Fehlererkennung hoch und die Wahrscheinlichkeit eines Falschalarms niedrig ist. Bei einer Ausführungsform decodiert das Verfahren mindestens einen Teil eines Steuerkanals, bestimmt, ob während der Decodierung ein oder mehrere Steuerkanäle erfolgreich empfangen wurden und wählt einen der sukzessiv empfangenen Steuerkanäle auf der Basis mindestens einer berechneten Metrik aus,

[0033] Genauer gesagt wird Teil I jedes gleichzeitig empfangenen HS-SCCH (d. h. 4 HS-SCCH) durch ein UE decodiert. Das UE enthält Techniken zum Bestimmen, ob ein oder mehrere der HS-SCCH während des Decodierungsschritts erfolgreich empfangen wurden. Wenn mehr als ein HS-SCCH erfolgreich empfangen wurde, wählt das Verfahren einen der sukzessiv empfangenen HS-SCCH auf der Basis berechneter Wegmetrikdifferenzen (PMD), die als Mechanismus zur „Gleichstandsauflösung“ dienen, um den korrekten HS-SCCH für ein bestimmtes UE auszuwählen.

[0034] Bei einer Ausführungsform verwendet das Verfahren einen Decodierungsalgorithmus des Typs Yamamoto-Itoh (YI), um zu bestimmen, wie viele HS-SCCH erfolgreich während des Decodierungsschritts empfangen wurden. Bei einer anderen Ausführungsform hängt eine Basisstation einen Paritätsprüfcode an Teil I jedes HS-SCCH an und das UE verwendet den Paritätsprüfcode während der Decodierung, um zu bestimmen, welche der HS-SCCH erfolgreich empfangen wurden. Der Paritätsprüfcode kann mit dem obenerwähnten YI-Algorithmus verwendet werden, um zusätzliche Zuverlässigkeit zu ergeben, und die berechneten PMD werden dann zum Auflösen etwaiger Gleichstände verwendet, falls mehrere HS-SCCH erfolgreich empfangen werden

sollten.

[0035] Bei einer weiteren Ausführungsform hängt eine Basisstation einen Code der Verwürfelungsgruppencodekennung (SCGI) an Teil I jedes HS-SCCH an und das UE prüft den angehängten SCGI-Code während der Decodierung, um zu bestimmen, welche der HS-SCCH erfolgreich empfangen wurden. Der SCGI-Code kann mit dem obenerwähnten YI-Algorithmus verwendet werden, um zusätzliche Zuverlässigkeit zu ergeben, und die berechneten PMD dienen dann zum Auflösen etwaiger Gleichstände, falls mehrere HS-SCCH erfolgreich empfangen werden sollten. Der YI-Algorithmus, der Paritätsprüfcodeansatz und der SCGI-Codeansatz können alle als „harte“ Metriken bezeichnet werden, die eine Bestimmung bezüglich „gut“ oder „schlecht“ an jedem HS-SCCH ergeben.

[0036] Die berechnete PMD-Metrik wird unter Verwendung mindestens einer von mehreren Metriken, die als „Soft“-Metriken bezeichnet werden können, berechnet. Die berechneten PMD-Metriken, die zur Auswahl nur eines HS-SCCH für ein UE verwendet werden, umfassen eine oder mehrere einer Metrik der minimalen Wegmetrikdifferenz (MPMD), einer Metrik der Aggregat-Wegmetrikdifferenz (APMD), einer Metrik der Häufigkeits-Wegmetrikdifferenz (FPMD) usw. Wie unten erläutert werden wird, wird jede dieser Metriken in Verbindung mit der Verwendung des obenerwähnten Viterbi-Decodierungsalgorithmus berechnet. Für den gewählten HS-SCCH decodiert das UE Teil II des HS-SCCH und beginnt mit dem Puffern von Daten aus dem gemeinsam benutzten Abwärtsstreckendatenkanal (HS-DSCH), der dem gewählten HS-SCCH entspricht.

KURZE BESCHREIBUNG DER ZEICHNUNGEN

[0037] Die vorliegende Erfindung wird aus der nachfolgend angegebenen ausführlichen Beschreibung und den beigefügten Zeichnungen, in denen gleiche Elemente durch gleiche Bezugszahlen repräsentiert werden und die lediglich als Veranschaulichung und nicht als Einschränkung der vorliegenden Erfindung angegeben werden, besser verständlich. Es zeigen:

[0038] [Fig. 1](#) einen Prozeßfluß zur Bildung einer typischen verketteten Codestruktur, die von einer Basisstation gemäß der Erfindung übertragen wird;

[0039] [Fig. 2](#) die Beziehung zwischen gemeinsam benutzten Steuerkanälen und gemeinsam benutzten Abwärtsstreckendatenkanälen gemäß der Erfindung;

[0040] [Fig. 3](#) die Verwendung eines Verwürfelungscodes mit einem Faltungsfehlerkorrekturcode;

[0041] [Fig. 4](#) ein Zustandsdiagramm eines einfachen Faltungscodierers;

[0042] [Fig. 5](#) ein Trellisdiagramm zur Erläuterung, wie der Viterbi-Algorithmus in einem Decoder eines UE implementiert wird;

[0043] [Fig. 6](#) ein Flußdiagramm des Fehlerdetektionsverfahrens gemäß der Erfindung;

[0044] [Fig. 7](#) ein Flußdiagramm einer Ausführungsform des Fehlerdetektionsverfahrens gemäß der Erfindung;

[0045] [Fig. 8](#) ein Trellisdiagramm zur Veranschaulichung, wie das Verfahren von [Fig. 7](#) erfolgreich den korrekten gemeinsam benutzten Steuerkanal decodiert; und

[0046] [Fig. 9](#) ein Flußdiagramm einer weiteren Ausführungsform des Fehlerdetektionsverfahrens gemäß der Erfindung; und

[0047] [Fig. 10](#) einen Prozeßfluß, wie Paritätsprüfcodebit in Verbindung mit dem Verwürfelungscode in der Basisstation verwendet werden.

AUSFÜHRLICHE BESCHREIBUNG

[0048] Obwohl sich die Prinzipien der Erfindung besonders gut für drahtlose Kommunikationssysteme auf der Basis der wohlbekannten HSDPA-Spezifikation (High Speed Downlink Packet Access) in dem UMTS-Standard (Universal Mobile Telecommunication System) eignen und in diesem beispielhaften Kontext beschrieben werden, versteht sich, daß die hier gezeigten und beschriebenen Ausführungsformen lediglich als Veranschaulichung und nicht als Einschränkung gedacht sind. Folglich werden Fachleuten verschiedene Modifikationen für die Anwendung auf andere Übertragungssysteme ersichtlich sein und werden durch die vorliegenden Lehren in Betracht gezogen. Im nachfolgenden Gebrauch ist außerdem Benutzergerät (UE) gleichbedeutend einer Mobilstation in einem drahtlosen Kommunikationsnetz.

[0049] Folglich wird nun ein Verfahren zum Erkennen von Fehlern in Steuerkanälen eines drahtlosen Kommunikationssystems beschrieben, das die Probleme der unzulänglichen Fehlerdetektion und der Falschalarmleistungsfähigkeit, die bei den oben beschriebenen herkömmlichen Verwürfelungs- oder EPMD-Ansätzen präsent sind, überwindet. Der herkömmliche Verwürfelungs- oder EPMD-Ansatz zur Fehlerdetektion ergibt gewöhnlich keine angemessene Fehlerdetektions-/Falschalarmleistungsfähigkeit. Der Grund dafür besteht darin, daß der Ansatz auf einer Wegmetrikdifferenz basiert, die nur auf der letzten Ebene in dem Trellis berechnet wird.

[0050] Die berechnete Metrik auf dieser letzten Ebene ist gewöhnlich eine sehr rauschbehaftete Metrik.

Der Grund dafür, daß EPMD eine rauschbehaftete Metrik ist (in bezug auf die Qualität eines Rahmens) besteht darin, daß EPMD nur einen letzten Wegmetrikvergleich verwendet, um zu bestimmen, ob ein Rahmen fehlerhaft ist oder nicht. EPMD berücksichtigt nicht, was während der „Reise“ des letztendlichen Gewinnerweges passiert, während er durch den Trellis voranschreitet. Folglich erfaßt der EPMD-Ansatz nicht den Umstand, daß auf einer früheren Ebene in der Decodierung (wie zum Beispiel in dem Trellisdiagramm von [Fig. 5](#) zu sehen ist) der Gewinnerweg möglicherweise einem anderen zusammenlaufenden oder konkurrierenden Weg sehr nahegekommen ist, wodurch er unzuverlässig wird.

[0051] [Fig. 6](#) ist ein Flußdiagramm des Fehlerdetektionsverfahrens gemäß der Erfindung. In [Fig. 6](#) empfängt ein UE zunächst bis zu vier (4) Teile I von 4 HS-SCCH über gleichzeitige Übertragungen (Schritt S10). Das UE entwürfelt Teil I in jedem HS-SCCH und decodiert jeden Teil I (Schritt S20). Gemäß dem Verfahren der Erfindung empfängt jede im folgenden beschriebene Ausführungsform verwürfelte Faltungscodes, die im Teil I jedes HS-SCCH enthalten sind. Die Verwürfelung geschieht durch den in [Fig. 3](#) skizzierten Verwürfelungsansatz. In dem UE invertiert das UE die Verwürfelung (d. h. entwürfelt die Bit) unter Verwendung seiner eigenen eindeutigen Sequenz wie oben beschrieben und leitet die entwürfelten Bit zur Decodierung zu einem Faltungsdecoder in dem UE. Der Schritt des Entwürfels wird von jeder der nachfolgend besprochenen Ausführungsformen ausgeführt.

[0052] Das Verfahren benutzt mehrere verschiedene Algorithmen oder Codes, um zu bestimmen (Schritt S30), wie viele HS-SCCH „gut“ oder „schlecht“ sind. Anders ausgedrückt, bestimmt dieser Schritt, ob während des Decodierungsschritts ein HS-SCCH erfolgreich empfangen wurde. Die verschiedenen Verfahren, die nicht als Einschränkung aufgefaßt werden sollten, werden im folgenden ausführlich beschrieben (siehe z.B. Schritt S130 in [Fig. 7](#) und Schritt S230 in [Fig. 9](#)). Im Schritt S40 wird die Anzahl erfolgreich empfangener HS-SCCH bestimmt. Wenn während der Decodierung nur ein HS-SCCH erfolgreich empfangen wurde (Ausgang von Schritt S40 ist JA), wird im Schritt S45 Teil II des HS-SCCH decodiert, und das UE beginnt mit dem Puffern von Daten aus dem HS-DSCH, der dem HS-SCCH entspricht.

[0053] Wenn entweder kein HS-SCCH ein Nachfolger war oder mehr als ein HS-SCCH erfolgreich empfangen wurde (NEIN im Schritt S40), bestimmt das UE, ob kein HS-SCCH erfolgreich empfangen wurde (Schritt S33). Wenn der Ausgang von Schritt S33 JA ist, wird die Verarbeitung angehalten (Schritt S34).

[0054] Wenn mehr als ein HS-SCCH ein Erfolg war

(NEIN im Schritt S33), das heißtt, daß zwei oder mehr HS-SCCH erfolgreich empfangen wurden, muß das Verfahren auf der Basis einer berechneten Metrik nur einen HS-SCCH auswählen (Schritt S50). Der Grund dafür besteht darin, daß es nur einen korrekten HS-SCCH für jedes UE geben kann, und daher nur einen HS-DSCH mit der beabsichtigten Übertragung für dieses UE. Im Schritt S50 wird für jeden verbleibenden guten HS-SCCH eine Wegmetrikdifferenz (PMD) berechnet. Die Wegmetrikdifferenz (PMD) ist die obenerwähnte Soft-Metrik, die als Gleichstandauflösungsmechanismus zwischen erfolgreich empfangenen HS-SCCH berechnet wird, um für die volle Decodierung nur einen HS-SCCH auszuwählen.

[0055] Zu berechneten PMD gehört die Berechnung einer minimalen Wegmetrikdifferenz (MPMD), einer Aggregat-Wegmetrikdifferenz (APMD) und einer Häufigkeits-Wegmetrikdifferenz (FPMD) oder Kombinationen einer oder mehrerer dieser Metriken, um nur einen HS-SCCH zur Decodierung von Teil II auszuwählen und zum Puffern von Daten aus dem dem gewählten HS-SCCH entsprechenden HS-DSCH. Wie später ausführlicher besprochen werden wird, werden die MPMD-, APMD- und FPMD-Metrikwerte jeweils in Verbindung mit der Verwendung des oben besprochenen Viterbi-Decodierungsalgorithmus bestimmt, um in jedem der Zustände einen Gewinnerweg zu bestimmen.

[0056] Diese Soft-Metriken aus dem Decodierungsprozeß werden im allgemeinen gegenüber Kanalqualitätsmetriken bevorzugt, da sie dem decodierten Bitstrom in der Empfängererkette im UE näher sind. Im Fall von Faltungscodes versuchen die Soft-Metriken, Differenzen in Wegmetriken zusammenlaufender Wege in jedem Zustand während der Viterbi-Decodierung auszunutzen. Folglich basieren die meisten der Soft-Metriken auf einem Decodierungsalgorithmus, der später ausführlicher erläutert werden wird. Bei der Beschreibung dieser Metriken werden die folgenden Notationen benutzt:

- (a) $\lambda_j(1)$ ist die Wegmetrik des gewählten (decodierten) Weges auf der j-ten Ebene in dem Trellis; und
- (b) $\lambda_j(2)$ ist die Wegmetrik des Weges, der mit dem gewählten Weg auf der j-ten Ebene in dem Trellis zusammenläuft.

[0057] Unter der Annahme, daß der als „1“ bezeichnete Weg der gewählte weg auf jeder Ebene ist, ist $\lambda_j(1)$ immer größer als $\lambda_j(2)$. Die tatsächlichen Werte der Wegmetriken sind im Vergleich zu ihren relativen Werten und der entsprechenden Anzahl decodierter Bitfehler nicht so wichtig.

[0058] Um die MPMD-Metrik zu berechnen, behält das Verfahren einen minimalen Wegmetrikdifferenzwert, um den der Gewinnerweg in jedem Zustand einen konkurrierenden oder zusammenlaufenden Weg

in jedem Zustand schlägt. Der Gewinnerweg mit dem größten minimalen Differenzmetrikwert entspricht dem HS-SCCH, der für die Decodierung von Teil II und die Pufferung der Daten auf dem entsprechenden HS-DSCH ausgewählt wird.

[0059] Um die MPMD-Metrik zu bestimmen, wird zusätzlich zu der Bestimmung des Gewinnerweges durch Viterbi-Decodierung die Metrik $\lambda_j(1) - \lambda_j(2)$ zur Zeit des Vergleichs behalten. wenn man annimmt, daß N Ebenen in dem Trellis vorliegen, wobei N die Anzahl der Informationsbit und Nachspannbit in einem Rahmen repräsentiert, ist die MPMD definiert als

$$\Lambda = \min_j [\lambda_j(1) - \lambda_j(2)]$$

wobei j der Ebenenindex ist. Am Ende der Viterbi-Decodierung wird der gewählte Weg ausgewertet und die minimale Metrik für diesen Weg wird als die Gewinnerwegmetrik oder Rahmenqualitätsmetrik gewählt. Die Intuition für die Verwendung der MPMD-Metrik besteht darin, daß, wenn auf irgendeiner Ebene während des Decodierungsprozesses ein zusammenlaufender Weg-Wert einem Gewinnerwegwert nahekommt, weniger Konfidenz in der Entscheidung der Auswahl dieses Gewinnerwegs besteht. Da der Gewinnerweg in dem Trellis während des Decodierungsprozesses offensichtlich nicht bekannt ist, wird während des Decodierungsprozesses der aktuelle minimale Wert in jedem Zustand gespeichert. Nachdem die Decodierung abgeschlossen ist, wird ein Array-Wert der Metrik entsprechend dem Abschlußzustand (durch Nachspannbit abgeschlossener Zustand) zu dem Wert der Gewinnerwegmetrik.

[0060] Für die APMD-Metrik behält das Verfahren eine Aggregatsumme der PMD, um die der Gewinnerweg in jedem Zustand einen konkurrierenden oder zusammenlaufenden Weg in jedem Zustand schlägt. Der Gewinnerweg mit der größten Aggregatsumme entspricht dem HS-SCCH, der für die Decodierung von Teil II und die Pufferung der Daten auf dem entsprechenden HS-DSCH ausgewählt wird.

[0061] Es wird die Summe der YI-Algorithmus-Metriken bei jedem Vergleich in dem Trellis ausgewertet. In der oben mit Bezug auf MPMD definierten Notation wird die APMD-Metrik gegeben durch

$$\Lambda = \sum_{j=1}^K [\lambda_j(1) - \lambda_j(2)]$$

[0062] Da die Anzahl der Ebenen in dem Trellis pro Rahmen festliegt, ist die aggregierte Metrik eine skalierte Version der mittleren Metrik. Die Intuition für die Verwendung der APMD-Metrik besteht darin, daß sie die Konfidenz repräsentiert, mit der der überlebende Gewinnerweg gegenüber dem zusammenlaufenden Weg in jedem Zustand in dem Trellis gewählt

wurde. Wie im Fall der minimalen Metrik ist ein Array mit einer Länge gleich der Anzahl der Zustände, die den aktuellen Wert von Λ für jeden Zustand enthalten, erforderlich. Am Ende des Decodierungsprozesses wird ein Arraywert der Metrik entsprechend dem Abschlußzustand (durch die Nachspannbit abgeschlossener Zustand) zu dem Wert der Gewinnerwegmetrik.

[0063] Für die FPMD-Metrik zählt das Verfahren, wie oft der Gewinnerweg in jedem Zustand innerhalb eines Schwellenwerts kam, einen konkurrierenden Weg in diesem Zustand zu schlagen. Der Gewinnerweg mit der niedrigsten Häufigkeit, innerhalb dieser Schwelle zu kommen, entspricht dem HS-SCCH, der für die Decodierung von Teil II und die Pufferung der Daten auf dem entsprechenden HS-DSCH ausgewählt wird.

[0064] [Fig. 7](#) ist ein Flußdiagramm einer Ausführungsform des Fehlerdetektionsverfahrens gemäß der Erfindung. Wie mit Bezug auf [Fig. 6](#) beschrieben, sind die Schritte S110, S120 und S150 in [Fig. 7](#) die selben wie die Schritte S10, S20 und S50 in [Fig. 6](#), so daß eine ausführliche Besprechung dieser Schritte weggelassen wird. Bei dieser Ausführungsform wird Teil I des HS-SCCH jedoch durch Verwendung des Codierungsalgorithmus von Yamamoto-Itoh (YI) codiert, und das Verfahren verwendet den Decodierungsalgorithmus von Yamamoto-Itoh (YI), um zu bestimmen (Schritt S130), wie viele HS-SCCH erfolgreich in dem UE empfangen wurden.

[0065] Der in der Literatur bekannte YI-Decodierungsalgorithmus bietet einen überlegenen Ansatz zur Fehlerdetektion in auf Rahmen basierende Kommunikation verwendenden Kommunikationssystemen. Auf jeder Ebene und in jedem Zustand in dem Trellisdiagramm von [Fig. 5](#) würde der YI-Algorithmus zum Beispiel ein Flag behalten, das angibt, ob der Gewinnerweg in diesem Zustand innerhalb einer Schwelle eines zusammenlaufenden Wegs gekommen ist. Wenn die Antwort JA ist, wird das Flag auf „unzuverlässig“ gesetzt; und wenn die Antwort NEIN ist, wird das Flag auf „zuverlässig“ gesetzt. Am Ende der Decodierung wird das Flag für den Gewinnerweg geprüft. Wenn ein Flag in dem Gewinnerweg auf „unzuverlässig“ gesetzt wurde, wird die Decodierung als erfolglos geblieben betrachtet. Im Kontext der Decodierung eines HS-SCCH bedeutet dies, daß das UE bestimmt, daß es nicht der beabsichtigte Empfänger ist.

[0066] Bei dem auf Verwürfelung basierenden Ansatz führt die Verwendung des YI-Algorithmus anstelle von EPMD zu einer viel besseren Detektions-/Falschalarmwahrscheinlichkeit. Im Kontext der Decodierung eines HS-SCCH ist der YI-Decodierungsalgorithmus alleine jedoch möglicherweise immer noch nicht ausreichend. Deshalb können verfei-

nernde Berechnungen von „Verbundmetriken“ erwünscht sein.

[0067] Zum Beispiel kann ein UE Datenteile von Teil I in jedem der 4 HS-SCCH decodieren und durch Verwendung des YI-Algorithmus von Schritt S130 bestimmen, daß mehr als einer der decodierten Teile I als zuverlässig ausgewertet wird.

[0068] Wenn nur ein HS-SCCH während der Decodierung erfolgreich empfangen wurde (Ausgang von Schritt S140 ist JA), wird im Schritt S145 Teil II des HS-SCCH decodiert, und das UE beginnt mit dem Puffern von Daten aus dem dem HS-SCCH entsprechenden HS-DSCH. Wenn entweder kein HS-SCCH ein Nachfolger war oder mehr als ein HS-SCCH erfolgreich empfangen wurden (NEIN in Schritt S140), bestimmt das UE, ob kein HS-SCCH erfolgreich empfangen wurde (Schritt S133). Wenn der Ausgang von Schritt S133 JA heißt, wird die Verarbeitung angehalten (Schritt S134).

[0069] Wenn mehr als ein HS-SCCH ein Erfolg war (NEIN im Schritt S133), bedeutet dies, daß zwei oder mehr HS-SCCH erfolgreich empfangen wurden, das Verfahren muß nur einen HS-SCCH auswählen (Schritt S150). Da es nur eine Übertragung zu einem UE pro TTI geben kann, ist offensichtlich einer dieser HS-SCCH ein Falschalarm. Gemäß dieser Ausführungsform der Erfindung wendet das UE folglich die oben mit Bezug auf die Schritte S50/S150 besprochene Prozedur der „Gleichstandsauflösung“ an, um den einen HS-SCCH für dieses UE auszuwählen.

[0070] [Fig. 8](#) ist ein teilweises Trellisdiagramm zur Darstellung der Funktionsweise des YI-Algorithmus. Der YI-Algorithmus kann als eine harte Metrik klassifiziert werden, die ein Gut-Ergebnis (z.B. die Übertragung ist für dieses UE bestimmt und das UE bestimmt erfolgreich, daß die Übertragung für es bestimmt ist), oder ein Schlecht-Ergebnis (z.B. das UE bestimmt, daß es nicht der beabsichtigte Empfänger ist). Wenn mehr als ein HS-SCCH als gut bewertet wird, wirken folglich die obenerwähnten Soft-Metriken als Gleichstandsauflösungsmechanismus, um den korrekten HS-SCCH und daher den korrekten HS-DSCH zu bestimmen. Die auf Wegmetrikdifferenzen basierenden MPMD-, APMD- und FPMD-Soft-Metriken können deshalb im Schritt S150 zur Verwendung als Faltungscodes berechnet werden, so daß das UE den korrekten HS-SCCH decodiert.

[0071] Für Anwendungen, bei denen die Verwendung eines CRC zur Fehlerdetektion kostspielig ist, liefert der YI-Algorithmus ein alternatives Verfahren zur Durchführung der Fehlerdetektion. Der Algorithmus arbeitet in Verbindung mit Viterbi-Decodierung von Faltungscodes mit sehr wenig Verarbeitungsverhead. Der YI-Decodierungsalgorithmus basiert

auf dem Prinzip, daß, wenn zwei Wege in dem Trellis zusammenlaufen und im Hinblick auf ihre Wegmetriken nahe beieinander sind, die Auswahl des einen Weges gegenüber dem anderen fehleranfälliger ist.

[0072] Der YI-Algorithmus wird im allgemeinen durch Betrachtung des Faltungscodes mit einfacher Rate [1/2] und der Nebenbedingungslänge $k = 3$ von [Fig. 4](#) erläutert, dessen Trellis in [Fig. 5](#) aufgetragen ist. Es folgt eine allgemeine Beschreibung des YI-Algorithmus. Als Anfang auf Ebene ($k - 1$) die $2k - 1$ Wege mit einem Label C identifizieren. Auf jeder der Ebenen j ($j = k, k + 1, \dots$) dann den Weg „a“ wählen, der den größten log-Wahrscheinlichkeitswert ($\lambda_j(a)$) aufweist, und den Weg „b“, der den nächstgrößten log-Wahrscheinlichkeitswert $\lambda_j(b)$ unter den zusammenlaufenden Wegen aufweist. Wenn ein Weg ein Label C auf Ebene ($j - 1$) aufweist und die Differenz zwischen $\lambda_j(a)$ und $\lambda_j(b)$ größer oder gleich A ist, einem Schwellenwert, der eine gegebene positive Konstante ist, den Weg „a“ mit dem Label C (z.B. zuverlässig) überleben lassen (überlebender Weg). Andernfalls überlebt Weg a mit dem Label X (z.B. unzuverlässig). Andere Wege als Weg „a“ werden in diesem Zustand verworfen.

[0073] Ein überlebender oder Gewinnerweg in jedem Zustand wird durch die Maximalwahrscheinlichkeitsentscheidung genauso ausgewählt, wie Wege unter Verwendung des gewöhnlichen Viterbi-Decodierungsalgorithmus ausgewählt werden. Der YI-Algorithmus reduziert sich auf den gewöhnlichen Viterbi-Decodierungsalgorithmus, wenn A, der Schwellenwert, gleich Null ist.

[0074] Nunmehr mit Bezug auf das teilweise Trellisdiagramm von [Fig. 8](#) nehme man an, daß die Wege a, b, c und d überlebende Wege mit Label C auf Ebene $j - 1$ sind. Auf der Ebene j laufen die Wege a-e und c-f in einem Zustand zusammen, und die Wege b-g und d-h in einem anderen. Im Fall

$$\lambda_j(a-e) \pm \lambda_j(c-f) + A$$

und

$$\lambda_j(d-h) < \lambda_j(b-g) = \lambda_j(d-h) + A$$

überlebt dann der Weg a-e mit dem Label C, und der Weg b-g überlebt mit dem Label X. Wie in [Fig. 8](#) gezeigt, laufen auf der Ebene $j + 1$ die Wege a-e-s und b-g-t zusammen. Auch im Fall

$$\lambda_{j+1}(b-g-t) < \lambda_{j+1}(a-e-s) + A$$

überlebt der Weg b-g-t mit dem Label X, weil der Weg b-g bereits auf Ebene j ein Label X aufweist.

[0075] Die obige Prozedur wird fortgesetzt, bis der gesamte Rahmen decodiert ist. An diesem Punkt

wird, wenn der überlebende Weg mit der besten Wegmetrik (der gewählte Weg) das Label X trägt, ein Fehler deklariert. Andernfalls wird der Rahmen als gut angenommen. Der YI-Algorithmus operiert mit der Intuition, daß, je weiter die Wegmetriken, die in einem Zustand verglichen werden, voneinander auseinander sind (Gewinnerweg zu anderem zusammenlaufenden Weg), desto größer die Konfidenz in dem gewählten überlebenden (Gewinner-)Weg ist.

[0076] [Fig. 9](#) ist ein Flußdiagramm einer weiteren Ausführungsform des Fehlerdetektionsverfahrens gemäß der Erfindung. Wie mit Bezug auf [Fig. 6](#) beschrieben wird, sind die Schritte S210, S220, S230 und S250 die selben wie die Schritte S10, S20, S40 und S50 in [Fig. 6](#), so daß eine ausführliche Besprechung dieser Schritte weggelassen wird. Bei dieser Ausführungsform verwendet das Verfahren jedoch einen Paritätsprüfcode, um zu bestimmen (im Schritt S230), wie viele HS-SCCH erfolgreich durch das UE empfangen wurden.

[0077] Ein weiterer Ansatz zur Fehlerdetektion besteht darin, einige wenige Paritätsprüfbit zu verwenden, um zusätzliche Zuverlässigkeit bei der Fehlerdetektion bereitzustellen. In einem Paritätsprüfcode wird jedes Paritätsprüfbit durch eine Exklusiv-OR-Operation an 2 oder mehr der Informationsbit berechnet.

[0078] Abhängig davon, wie jedes Paritätsprüfbit berechnet wird, ergeben sich für dieselbe Anzahl von Paritätsbit vielfältige Paritätsprüfcodes. Wenn man zum Beispiel annimmt, daß drei Informationsbit und zwei Paritätsprüfbit vorliegen, läge ein möglicher Paritätsprüfcode vor, wenn das erste Paritätsprüfbit durch Exklusiv-OR des 1. und 3. Informationsbit und das zweite Paritätsprüfbit durch Exklusiv-OR des 2. und 3. Bit berechnet wird. Ein anderer Paritätsprüfcode kann das erste Paritätsbit unter Verwendung des 1. und 2. Informationsbit berechnen und das 2. Paritätsbit wird unter Verwendung des 1. und 3. Informationsbit berechnet. Ein oder mehrere Paritätsbit können auch einfach Wiederholungen bestimmter Informationsbit sein.

[0079] Das Paritätsprüfergebnis verbessert, wenn es in Verbindung mit dem YI-Algorithmus und den oben definierten PMD-Soft-Metriken verwendet wird, die Detektions-/Falschalarmleistungsfähigkeit sehr. [Fig. 10](#) zeigt einen Prozeßfluß, der veranschaulicht, wie Paritätsprüfcodebit in Verbindung mit dem Verwürfelungscode in der Basisstation verwendet werden. Man betrachte zum Beispiel einen Teil I eines HS-SCCH mit 8 Informationsbit, wobei 4 Paritätsprüfbit berechnet und an die 8 Informationsbit in Block **1010** angehängt werden. Für jeden HS-SCCH werden die Informationsbit mit angehängten Paritätsbit im Block **1020** faltungscodiert und dann im Block **1030** unter Verwendung des mit Bezug auf [Fig. 3](#) be-

schriebenen Verwürfungsansatzes verwürfelt. Bei 4 HS-SCCH vergrößert dies die Spitzendatenratenverarbeitung auf 72 Kbps, was immer noch wesentlich besser als das bei Verwendung eines 12-Bit- oder höheren CRC-Codes erforderliche ist.

[0080] Nunmehr mit Bezug auf [Fig. 9](#) kann ein UE Datenteile von Teil I in jedem der 4 HS-SCCH decodieren und im Schritt S230 durch Verwenden des Paritätsprüfcodes bestimmen, daß einer oder mehrere der decodierten Teile I als zuverlässig ausgewertet sind. Genauer gesagt wird mit dem durch das UE decodierten Paritätsprüfcodes geprüft, ob etwaige der HS-SCCH bestehen. Wenn ein SCCH ein Erfolg war (JA im Schritt **231**), wird dann wie bei der vorherigen Ausführungsform Schritt S245 durchgeführt: Decodieren von Teil II für diesen HS-SCCH und Beginnen mit dem Puffern von Daten aus dem entsprechenden HS-DSCH. Wenn entweder kein HS-SCCH ein Nachfolger war oder mehr als ein HS-SCCH erfolgreich empfangen wurde (NEIN in Schritt S231), bestimmt das UE, ob kein HS-SCCH erfolgreich empfangen wurde (Schritt S233). Wenn der Ausgang von Schritt S233 JA ist, wird die Verarbeitung angehalten (Schritt S234). Wenn mehr als ein HS-SCCH ein Erfolg war (NEIN im Schritt S233), verwendet das Verfahren in einem Schritt S235 den YI-Algorithmus, um einen Zuverlässig- oder Unzuverlässig-Indikator der HS-SCCH zu erhalten und die Verarbeitung kehrt zum Schritt S231 zurück. Wenn im Schritt S231 genau ein SCCH ein Erfolg war (Ausgang von Schritt S231 ist JA), dann im Schritt S245 Decodierung Teil II für diesen HS-SCCH und Beginnen mit dem Puffern von Daten aus dem entsprechenden HS-DSCH. Wenn keine HS-SCCH bestanden haben oder wenn mehr als ein HS-SCCH immer noch bestanden hat (Ausgang von Schritt S233 ist NEIN bei der zweiten Iteration, durch gestrichelte Linien dargestellt), werden dann im Schritt S250 die Metriken APMD, FPMD und/oder MPMD berechnet und zum Auflösen von Gleichständen verwendet, um so nur einen HS-SCCH zur Decodierung auszuwählen.

[0081] Da es nur eine Übertragung zu einem UE pro TTI geben kann, ist offensichtlich einer der HS-SCCH ein Falschalarm. Gemäß dieser Ausführungsform der Erfindung kann das UE somit letztendlich die anfänglich oben im Schritt S50 von [Fig. 6](#) besprochene Prozedur der „Gleichstandsauflösung“ anwenden, um den einen HS-SCCH für dieses UE auszuwählen.

[0082] Bei einer weiteren Ausführungsform wird die Fehlerdetektion unter Verwendung eines alternativen Codes zu dem Paritätsprüfcodes durchgeführt, der als Code der Verwürfungscodegruppenkennung (SCGI) bezeichnet wird. Der SCGI-Code wird an jeden Teil 1 des HS-SCCH wie in [Fig. 10](#) angehängt. Die SCGI identifiziert die Verwürfungscodegruppe, zu der die HS-SCCH-Übertragung gehört, und wird zusammen mit Teil I übertragen. Jedes UE decodiert

Teil I auf jedem HS-SCCH und prüft, ob die SCGI seiner UE-spezifischen Verwürfungscodegruppe entspricht. Wenn keiner der SCGI-Codes mit der UE-spezifischen Verwürfungscodegruppe übereinstimmt, bestimmt das UE, daß keiner der HS-SCCH für es bestimmt sind. Wenn genau ein SCGI-Code übereinstimmt, schreitet es mit der Decodierung von Teil II dieses SCCH voran und beginnt außerdem mit dem Puffern des entsprechenden DSCH. Wenn mehr als ein SCGI-Code mit seiner Gruppe übereinstimmt, löst das UE den Gleichstand dann entweder unter Verwendung des YI-Algorithmus und/oder der Soft-Metriken wie oben definiert auf.

[0083] Das Prinzip hinter dem obigen Ansatz besteht darin, die Gesamtmenge von Verwürfungs-codes in Gruppen aufzuteilen. Wenn eine p-Bit-SCGI verwendet wird, werden 2^p Gruppen gebildet. Die Vergabe von Verwürfungs-codes an UEs (die zum Verbindungsaufbauzeitpunkt geschieht) gleicht die Anzahl der UEs in jeder Verwürfungscodegruppe aus. Dies verringert die Wahrscheinlichkeit gleichzeitiger Übertragungen zu UEs in derselben Codegruppe, wodurch die Falschalarmwahrscheinlichkeit verbessert wird. Der SCGI-Ansatz ist in der Basisstation und in dem UE relativ leicht zu implementieren, da im Gegensatz zu dem Paritätsbitcode lediglich erforderlich ist, das SCGI-Bitfeld vor der Codierung von Teil I eines HS-SCCH in die Informationsbit einzufügen oder daran anzuhängen.

[0084] Nachdem die Erfindung somit beschrieben wurde, ist offensichtlich, daß sie auf vielerlei Weise variiert werden kann. Die oben beschriebenen Algorithmen wurden als aus mehreren Komponenten, Flußdiagrammen oder Blöcken bestehend beschrieben, es versteht sich jedoch, daß die Verfahren in anwendungsspezifischen integrierten Schaltungen, softwaregesteuerten Prozessorschaltkreisen oder in anderen Anordnungen diskreter Komponenten implementiert werden können.

Patentansprüche

1. Verfahren zum Auswählen eines Steuerkanals aus mehreren Steuerkanälen (**210**), die gleichzeitig oder parallel in einem drahtlosen Kommunikationssystem von einer Basisstation zu mehreren Benutzergeräten UE gesendet werden, wobei jeder Steuerkanal (**210**) mit Faltungscodes codiert und durch verschiedene Verwürfungs-codes verwürfelt wird und einen ersten Teil Teil 1 und einen zweiten Teil Teil 2 aufweist, gekennzeichnet durch die folgenden Schritte:

Decodieren des ersten Teils jedes Steuerkanals in einem Benutzergerät unter Verwendung eines Viterbi-Decoders oder Viterbi-Algorithmus, wobei der erste Teil eine Verwürfungscodegruppenkennung SCGI enthält, die an diesen angehängt wird; Bestimmen, ob während des Decodierungsschritts

ein oder mehrere Steuerkanäle erfolgreich empfangen wurden, durch Prüfen, ob der angehängte SCGI-Code einer Verwürfelungscodegruppe des UE entspricht; und

Auswählen eines der erfolgreich empfangenen Steuerkanäle auf der Basis mindestens einer berechneten Metrik, wobei die mindestens eine berechnete Metrik mindestens eine Wegmetrikdifferenz PMD umfaßt.

2. Verfahren nach Anspruch 1, wobei der Schritt des Bestimmens den folgenden Schritt umfaßt:

Ausführen eines Yamamoto-Itoh-YI-Decodierungsalgorithmus in Verbindung mit dem Prüfen, ob der angehängte SCGI-Code einer Verwürfelungscodegruppe des UE entspricht, um zu bestimmen, ob während des Decodierungsschritts ein oder mehrere Steuerkanäle erfolgreich empfangen wurden.

3. Verfahren nach Anspruch 2, wobei der Schritt des Auswählens die folgenden Schritte umfaßt:

Berechnen einer PMD-Metrik der Wegmetrikdifferenz für jeden Steuerkanal, wenn mehr als ein Steuerkanal erfolgreich empfangen wurde; und

Auswählen eines der erfolgreich empfangenen Steuerkanäle auf der Basis der berechneten PMDs.

4. Verfahren nach Anspruch 3, wobei der Schritt des Berechnens den folgenden Schritt umfaßt:

Berechnen einer MPMD-Metrik der minimalen Wegmetrikdifferenz, einer APMD-Metrik der Aggregat-Wegmetrikdifferenz und/oder einer FPMD-Metrik der Häufigkeits-Wegmetrikdifferenz.

5. Verfahren nach Anspruch 4, wobei der Schritt des Berechnens einer MPMD-Metrik den folgenden Schritt umfaßt:

Ausführen einer Viterbi-Decodierung, um in jedem von mehreren Zuständen eine Gewinner-Wegmetrik zu bestimmen, und

Behalten eines minimalen Wegdifferenzwerts, um den der Gewinnerweg in jedem Zustand einen konkurrierenden Weg in jedem Zustand schlägt, wobei der Gewinnerweg mit dem größten minimalen Wegdifferenzmetrikkwert dem in dem Schritt des Auswählens ausgewählten erfolgreich empfangenen Steuerkanal entspricht.

6. Verfahren nach Anspruch 4, wobei der Schritt des Berechnens einer APMD-Metrik den folgenden Schritt umfaßt: Ausführen einer Viterbi-Decodierung,

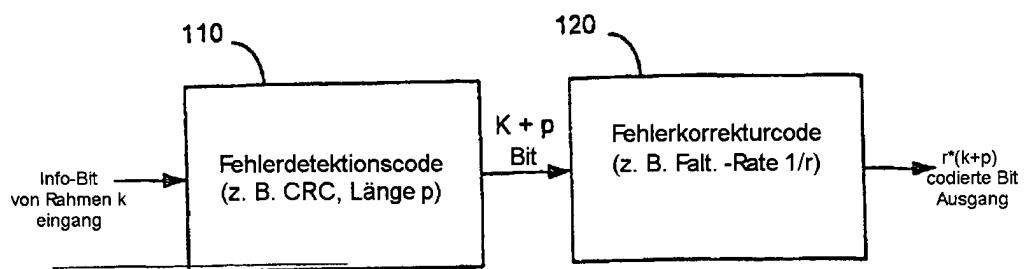
um in jedem von mehreren Zuständen eine Gewinner-Wegmetrik zu bestimmen, und Behalten einer Aggregatsumme der PMDs, um die der Gewinnerweg in jedem Zustand einen konkurrierenden Weg in jedem Zustand schlägt, wobei der Gewinnerweg mit der größten Aggregatsumme dem in dem Schritt des Auswählens ausgewählten erfolgreich empfangenen Steuerkanal entspricht.

7. Verfahren nach Anspruch 4, wobei der Schritt des Berechnens einer FPMD-Metrik den folgenden Schritt umfaßt: Ausführen einer Viterbi-Decodierung, um in jedem von mehreren Zuständen eine Gewinner-Wegmetrik zu bestimmen; und Zählen einer Anzahl von Malen, um die der Gewinnerweg in jedem Zustand innerhalb eines Schwellenwerts kam, einen konkurrierenden Weg in jedem Zustand zu schlagen, wobei der Gewinnerweg mit der niedrigsten Häufigkeit, innerhalb der Schwelle zu kommen, dem in dem Schritt des Auswählens ausgewählten erfolgreich empfangenen Steuerkanal entspricht.

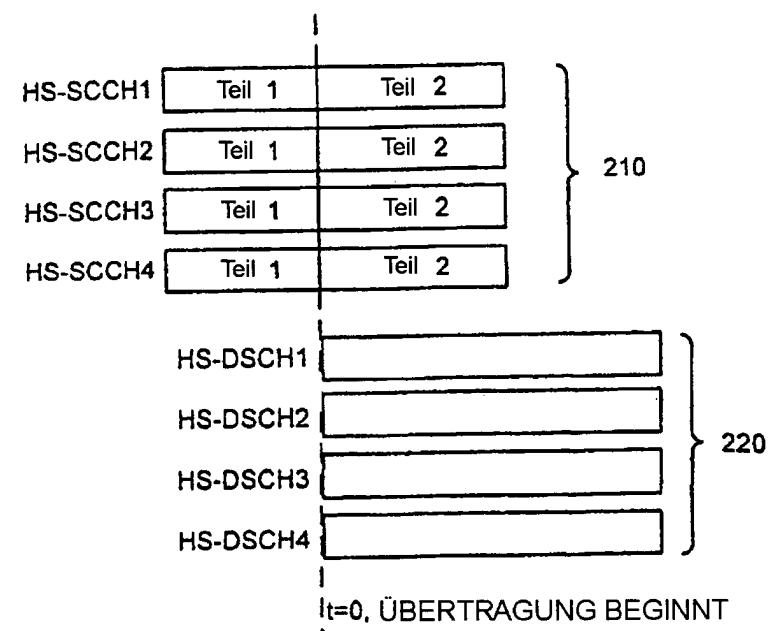
Es folgen 6 Blatt Zeichnungen

Anhängende Zeichnungen

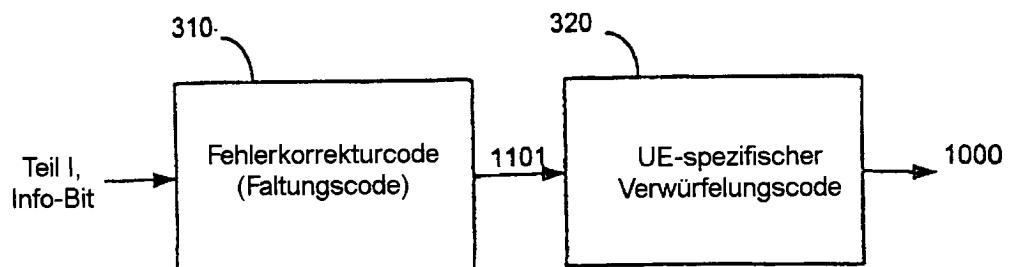
Figur 1



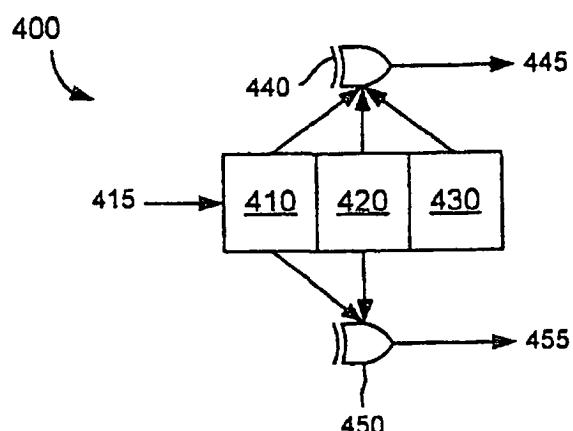
Figur 2



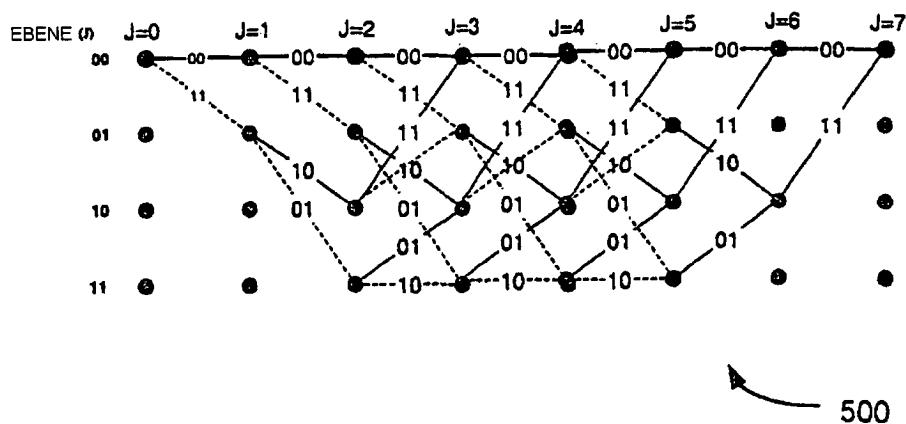
Figur 3



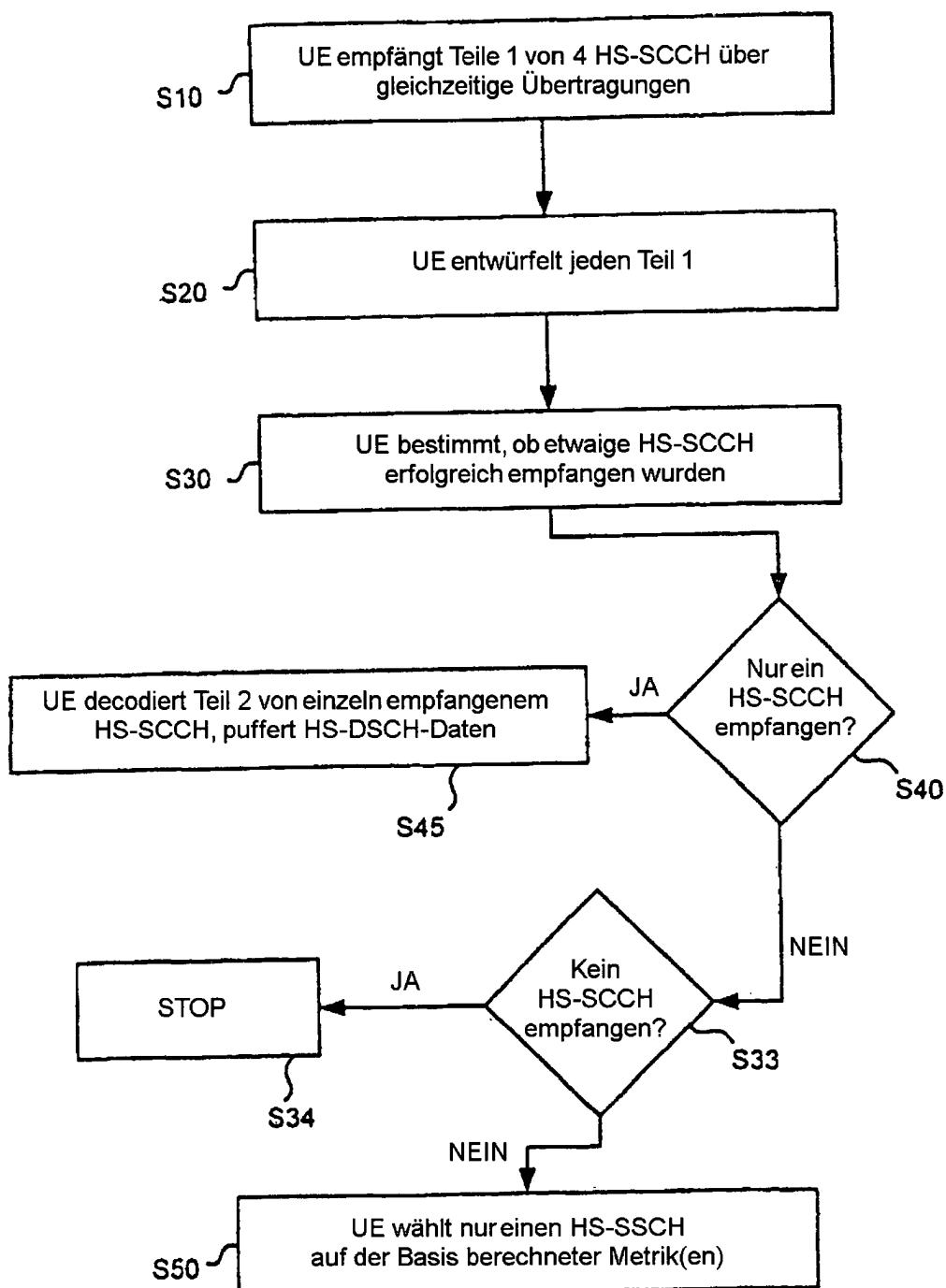
Figur 4



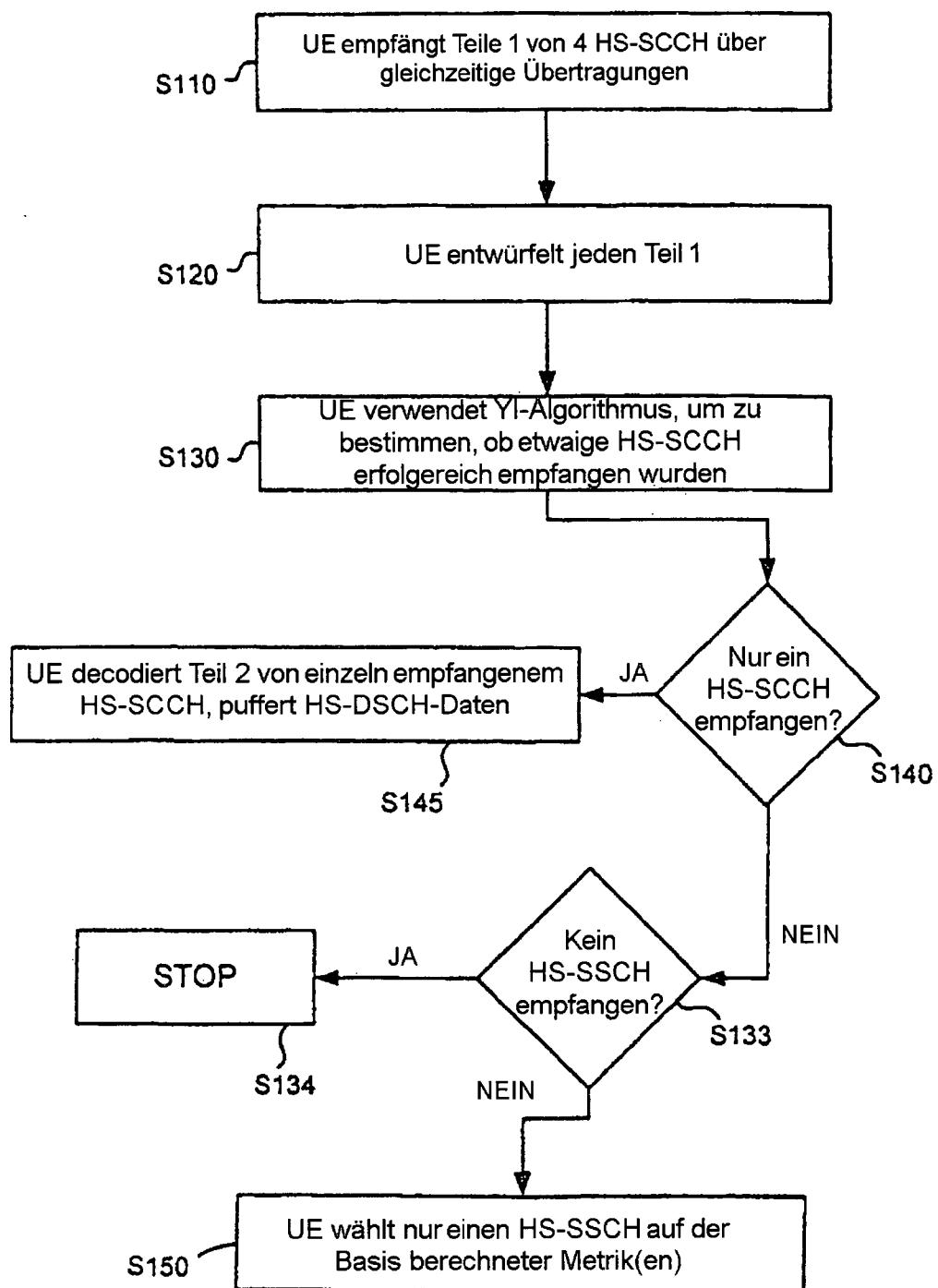
Figur 5



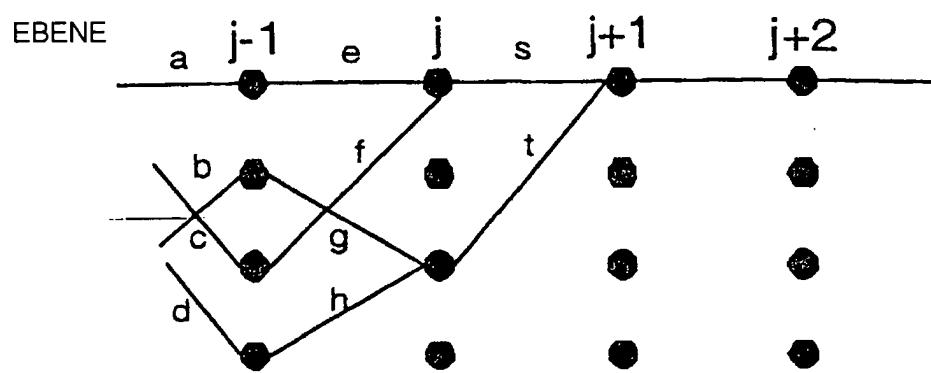
Figur 6



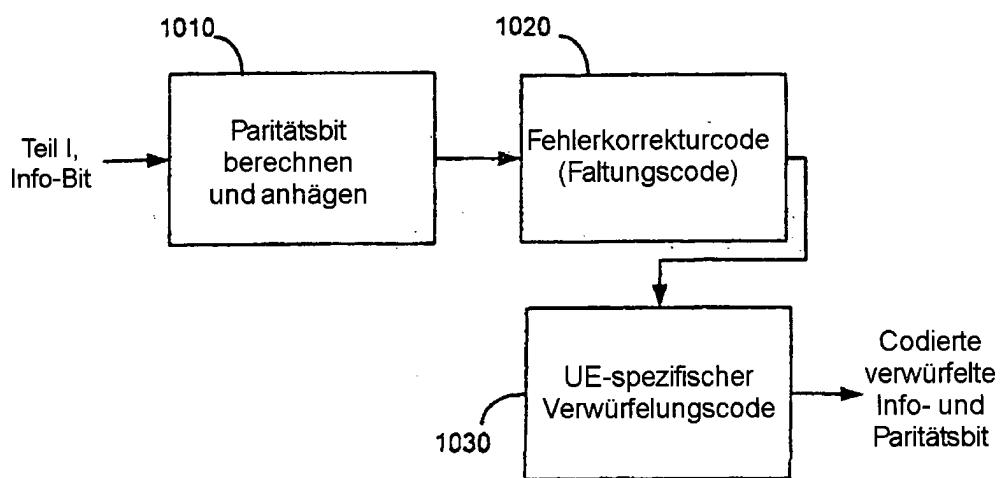
Figur 7



Figur 8



Figur 10



Figur 9

