Lehrstuhl für Nachrichtentechnik

Turbo-Detection for GSM-Systems - Channel Estimation, Equalization and Decoding

Dipl.-Ing. (Univ.) Volker Franz

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs

genehmigten Dissertation.

Vorsitzender: Prof. Dr.-Ing. J. Ebersächer

Prüfer der Dissertation: 1. Univ.-Prof. Dr.-Ing. J. Hagenauer

2. Univ. Prof. Dr.-Ing. M. Bossert
Universität Ulm

Die Dissertation wurde am 19.5.2000 bei der Technischen Universität München eingereicht und durch die Fakultät for Elektrotechnik und Informationstechnik am 11.10.2000 angenommen

Vorwort

Die vorliegende Arbeit entstand in der Zeit von August 1996 bis Dezember 1999 im Rahmen meiner Tätigkeit als Doktorand bei der Siemens AG in München und bei Prof. Dr.-Ing. J. Hagenauer am Lehrstuhl für Nachrichtentechnik der Technischen Universität München. Ich möchte all jenen danken, die mich während der Entstehung dieser Arbeit unterstützt haben.

Mein besonderer Dank ergeht an Prof. J. Hagenauer für die Anregung, die Betreuung und die Förderung meiner Arbeit. Durch seine Diskussionsbereitschaft sowie durch zahlreiche Ratschläge und Hinweise hat er wesentlich zum Gelingen dieser Arbeit beigetragen.

Herrn Prof. Dr.-Ing. M. Bossert danke ich für das Interesse an dieser Arbeit und für die Übernahme des Koreferats. Weiterhin danke ich dem Vorsitzenden der Prüfungs-kommision, Prof. Dr.-Ing. J. Eberspächer.

Besonders möchte ich mich bei der Siemens AG, vor allem bei Herrn Dr.-Ing. J. Sokat und Herrn Dr. Fuchs, dafür bedanken, dass mir die vorliegende Arbeit ermöglicht wurde. Desweiteren möchte ich mich bei Herrn Dipl.-Ing. D. Emmer für sein Engagement bei der Betreuung dieser Arbeit danken. Herrn Dr.-Ing. L. Rademacher und Herrn Dipl.-Ing. E. Humburg danke ich für die zahlreichen Diskussionen, sowie der gesamten ER 5 für die angenehme Arbeitsatmosphäre. Bei meiner Diplomandin Han Zhang bedanke ich mich für ihren Beitrag in dieser Arbeit.

Mein besonderer Dank gilt Herrn Dipl.-Ing G. Bauch, Lehrstuhl für Nachrichtentechnik, Technische Universität München, für die erfolgreiche und harmonische Zusammenarbeit.

Bedanken möchte ich mich bei meiner Familie, meiner Freundin Jasmin, bei Roland Morasch und bei meinen Freunden, die mich während meiner Doktorarbeit immer uneingeschränkt unterstützt und ausgehalten haben.

Volker Franz

ContentsI

Abstract

In this work the benefits and the limits of turbo-detection for GSM-systems are examined.

Turbo-detection applies the turbo-principle to iterative equalization and decoding. The channel encoder and the transmission channel are regarded as a serial concatenation of convolutional codes, and hence, can be iteratively decoded. Since turbo-detection requires modifications of the transmission only at the receiver, it can be adopted to existing mobile radio systems without any amendment of the transmission standard.

Before applying this method to existing GSM-services, e.g. full-rate speech, the original turbo-detection scheme must be changed to enable acceptable transmission delays. New schemes are developed that differ in complexity, delay, memory requirement, and performance. The performance difference is evaluated using time-invariant intersymbol-interference channels. It is shown that the performance of the original turbo-detection scheme can be maintained for these channels. The influence of channel parameter estimation and suboptimum detection algorithms is investigated.

However, in order to evaluate the benefits for GSM-systems, the transmission channel has to comprise an exact model of the modulator and the mobile radio channel. Applying the turbo-detection schemes to GMSK modulated GSM-services, only small iteration gains are obtained after decoding due to the orthogonal intersymbol-interference of the modulator. Even for multipath environments with large delay spreads the iterations gains do not exceed 0.7 dB since the paths are fading independently.

Higher-order modulation schemes will be used for new high-data rate services within GSM. Introducing the symbol soft-values, the turbo-detection scheme can be easily adopted to these modulation techniques. It is shown that for the new packet switched services large iteration gains of up to 2 dB are achieved. By considering the signal-to-noise distribution in the network, the throughput of the system, and hence, the spectral efficiency can be improved by up to 30%. Simulation results show that turbo-detection also guarantees large gains in severe mobile radio environments. Additionally, turbo-detection is applied to a system of serial concatenated convolutional codes. By iteratively detecting this double serial concatenated system the performance is further increased compared to simple turbo-decoding.

For fast-fading channels an adaptive Max-Log-MAP equalizer is developed that can be incorporated in the turbo-loop. This adaptive equalizer profits from the decoder decisions and accurately tracks the channel variations. However, no significant iteration gain is achieved after decoding for GMSK-modulated services.

Zusammenfassung

Einen der am schnellsten wachsenden Telekommunikationsmärkte der letzten Jahrzehnte stellt die mobile Kommunikation dar. Der Durchbruch dieser Technologie begann in den frühen achtziger Jahren mit der Einführung der zellularen Mobilfunknetze der "ersten Generation". Vertreter dieser Generation sind AMPS, TACS, NMT, C-450, etc. Charakteristisch für diese Netze sind analoge Modulationsverfahren. Direkt nach der Einführung dieser Systeme wurde deutlich, dass selbst diese neuen analogen Systeme schnell an ihre Grenzen stoßen. Wesentliche Einschränkung stellt zum einen die Konzeption dieser Systeme für eine geringe Anzahl an Benutzern dar. Die Nachfrage nach mobiler Telephonie war stark unterschätzt worden. Somit war es schon zur Einführung der Systeme klar, dass die Kapazitätsgrenzen der Systeme in kürzester Zeit ausgereizt werden. Zum anderen sind die Systeme zueinander nicht kompatibel, was eine länderübergreifende Nutzung desselben Telefons unmöglich macht. So wurde schon 1982 die "Groupe Spéciale Mobile" (GSM, später: global system for mobile communications) innerhalb der CEPT (conférence européene des postes et télécommunications) gegründet. Ziel dieser Gruppe war es, einen neuen Mobilfunkstandard zu entwickeln, der die Nachteile der "ersten Generation" vermeiden sollte und den Anforderungen des schnell wachsenden Marktes der mobilen Kommunikation gerecht würde. Hierzu wurden bandbreiteneffiziente Übertragungsverfahren untersucht und die Integration neuer, vielfältiger Dienste in Betracht gezogen. 1990 wurde der GSM-Standard der Phase 1 verabschiedet und 1995 durch die Phase 2 ergänzt. Sowohl im GSM-Standard als auch in den anderen Mobilfunkstandards der "zweiten Generation" werden digitale Übertragungstechniken eingesetzt. Weitere Vertreter der "zweiten Generation" sind D-AMPS in USA und PDC in Japan. Seit der Einführung des ersten kommerziellen GSM-Netzes stieg die Anzahl der Teilnehmer von ca. einer Million im Jahre 1993 auf ca. 138 Millionen im Jahre 1998. Dabei ist eine Stagnation dieses Wachstumsprozesses noch nicht abzusehen.

Heute haben schon zahlreiche Netze ihre Kapazitätsgrenzen erreicht und die zugewiesenen Frequenzen reichen für den steigenden Bedarf an Mobiltelephonie nicht mehr aus. Zusätzlich wird die Nachfrage nach mehr Übertragungskapazität durch die ständig wachsende Steigerungsrate der Datenübertragung beschleunigt. Um diesen Anforderungen gerecht zu werden, werden innerhalb des GSM-Standards der Phase 2+ neue Dienste definiert. So sollen die Datenraten von Diensten wie GPRS (general packet radio services) und HSCSD (high-speed circuit switched data) innerhalb von EDGE (enhanced data rates for GSM evolution) mit Hilfe von höherstufigen Modulationsverfahren wie der 8-PSK-Modulation gesteigert werden. Dabei werden die modulierten Signale störanfälliger und die spektrale Effizienz kann nicht verdreifacht werden. Zur weiteren Steigerung der Kapazität und Effizienz von GSM-Systemen ist die Anwendung von modernen Techniken der Informationstheorie und der Codierungstheorie notwendig.

Die Geburt der Informationstheorie wird mit der Arbeit von C. E. Shannon [Sha48] im Jahre 1948 verbunden. In dieser und in späteren Arbeiten wie "Communications in the Presence of Noise" [Sha49] zeigte C. E. Shannon, dass fehlerfreie Übertragung dann mit einer bestimmten Signalenergie möglich ist, wenn die Datenrate unterhalb einer sogenannten Shannon-Grenze liegt. Leider konnte Shannon nicht zeigen, wie eine solche Übertragung in der Praxis effizient gestaltet werden kann und wie eine entsprechende, realisierbare Kanalcodierung aussieht.

Da Shannon auch nachgewiesen hat, dass die Übertragungsqualität durch eine Vergrößerung der Codelänge verbessert werden kann, lag zunächst das Hauptinteresse der Forscher darin, starke Codes mit großer Blocklänge zu finden. Die Komplexität der Decodierung dieser Codes war zu hoch für eine Implementierung. Eine neue Strategie lag in der Aufteilung der langen Codes in kleinere, verkettete Codes [For66]. Mit diesen Codes wächst die Decodierkomplexität nicht exponentiell mit der Länge des Codes. Außerdem erlaubt der schnelle Fortschritt in der Halbleiterindustrie die Realisierung komplexer Algorithmen. Neben den harten Entscheidungen können auch noch Zuverlässigkeitswerte zwischen einzelnen Decodern ausgetauscht werden [Hag94]. Diese Module werden SISO-(soft-in/soft-out)-Module genannt. Sie verwenden die Zuverlässigkeitsinformationen vorheriger Decodierstufen und generieren wiederum Zuverlässigkeitsinformationen. Durch die Verwendung der Zuverlässigkeitsinformation wird die Performanz der Decodierung stark gesteigert.

Ein weiterer Meilenstein in der Codierungstheorie sind die sogenannten Turbo-Codes. Sie wurden 1993 von Berrou et al. [BGT93] vorgestellt. Basierend auf dem Prinzip der iterativen Decodierung [BDG79, LYH93], wurde eine iterative Decodierung von parallel verketteten Faltungscodes vorgeschlagen. Mit dieser Technik konnte man der Shannon-Grenze ein weiteres Stück näher kommen. So wurden in den folgenden Jahren die Eigenschaften von parallel und seriell verketteten Faltungs- und Blockcodes genauer untersucht [BDM98b,BeG96, BeM96, HOP96].

Ein großes Interesse in der modernen Codierungstheorie liegt in der Anwendung des sogenannten Turbo-Prinzips [Hag97] auf andere Detektionsverfahren. Eine typische Applikation hierfür ist die iterative Entzerrung und Decodierung, die auch Turbo-Detektion genannt wird. Zuerst veröffentlicht in [DJB95] folgten weitere Arbeiten in [BKH97, PDG97, GLL97].

Der Kanalcoder und der Übertragungskanal können als ein seriell verkettetes Codesystem betrachtet werden, das iterativ decodierbar ist. Um die Turbo-Detektion anwenden zu können, muss lediglich der Empfänger angepasst werden und keine Änderung des Mobilfunkstandards erfolgen. Somit stellt die Turbo-Detektion eine Möglichkeit dar, die Kapazität existierender und zukünftiger Mobilfunksysteme zu steigern. Bei einer Verbesserung der Empfangssensitivität des Empfängers kann der Netzbetreiber in zweierlei Hinsicht profitieren. Er kann bei gleicher Qualität den Zellradius vergrößern, die Anzahl der Antennen und die damit verbundenen Kosten reduzieren. Im Falle schon vorhandener Infrastruktur kann die Anzahl der bedienbaren Teilnehmer gesteigert werden.

Diese Arbeit untersucht das Verhalten der Turbo-Detektion für existierende GSM-Systeme. Zusätzlich werden neue Dienste, die auf höherstufigen Modulationsverfahren basieren, betrachtet. Potentiale und Grenzen der Turbo-Detektion werden aufgezeigt.

Viele GSM-Dienste verwenden ein sogenanntes Interblock-Interleaving. So auch der wichtigste Dienst, der vollratige Sprachkanal TCH/FS (traffic channel/full-rate speech). Bei diesem Interleaving werden mehrere Codeblöcke miteinander vermischt und übertragen. Eine Modifikation des herkömmlichen Turbo-Detektionsschemas ist für eine Implementierung essentiell, da die Übertragungsverzögerung sonst unendlich groß wird. In dieser Arbeit werden mehrere Detektionsschemata vorgestellt und mit dem ursprünglichen Verfahren bzgl. des Fehlerverhaltens, der Komplexität, der Verzögerung und des Speicherbedarfs verglichen.

Anhand von zeitinvarianten Intersymbol-Interferenz-(ISI)-Kanälen wird gezeigt, dass durch relativ einfache Verarbeitungsschemata hohe Iterationsgewinne erreichbar sind. Durch zusätzliche Verzögerungen bezogen auf nicht-iterative Verfahren, kann die Performanz der ursprünglichen Turbo-Detektion angenähert werden. Die Einflüsse des Interblock-Interleaving auf die Turbo-Detektion werden somit stark reduziert. Untersuchungen zu dem Einfluss der Kanalparameterschätzung zeigen, dass die Iterationsgewinne durch die nicht-ideale Kanalkenntnis geringfügig sinken. Außerdem wird durch Simulationen nachgewiesen, dass mit suboptimalen Empfängern wie dem Max-Log-MAP-Algorithmus ähnliche Iterationsgewinne erzielbar sind. Der Max-Log-MAP-Algorithmus ist auf Grund seiner reduzierten Komplexität, seiner Stabilität und seines nahezu optimalen Verhaltens für eine Implementierung in realen Systemen zu empfehlen.

Der zeitinvariante Kanal ist für einen Vergleich der verschiedenen Detektionsverfahren geeignet. Um jedoch den Nutzen der Turbo-Detektion für GSM-Systeme genauer bewerten zu können, sollte ein Kanal verwendet werden, der die Realität ausreichend abbildet. Der Kanal muss dabei den in GSM-Systemen verwendeten GMSK-Modulator und den zeitvarianten Mobilfunkkanal modellieren. Hier werden nur geringe Gewinne für die Blockfehlerraten und die Bitfehlerraten nach der Decodierung erzielt. Die Iterationsgewinne nach dem Entzerrer für einen TU50-Kanal mit idealem Frequenzsprungverfahren sind größer als 2dB. Bei der Untersuchung des Übertragungskanals wird gezeigt, dass das Ausbleiben von Iterationsgewinnen nicht auf die Zeitvarianz der Kanäle sondern auf die Beschaffenheit des Modulators zurückzuführen ist. Eine wesentliche Eigenschaft des GMSK-Modulators ist, dass die Impulsantworten von zwei aufeinanderfolgenden Bits orthogonal zueinander sind. Die ausschlaggebende Interferenz verursacht das übernächste Bit. Sie ist jedoch so gering, dass die Iterationsgewinne nach dem Decoder nicht signifikant sind. Größere Gewinne können erzielt werden, wenn der Delay-Spread des Kanals größer ist. Dies trifft z.B. für eine Übertragung mit Verzögerungsdiversität zu. Die hier erzielten Iterationsgewinne bleiben mit ca. 0.7 dB gering.

In den oben genannten Untersuchungen wurde eine nicht-adaptive Kanalschätzung verwendet. Besonders bei sich schnell verändernden Übertragungskanälen degradiert das Fehlerverhalten, wenn die Kanalschätzung die zeitlichen Änderungen nicht kompensieren kann. Die hierzu verwendeten adaptiven Kanalschätzverfahren basieren auf den Entzerrerentscheidungen. Da bei der Turbo-Detektion für GMSK-basierte Dienste diese Entscheidungen signifikant verbessert werden können, kann die Einbeziehung einer adaptiven Kanalschätzung mit rückgekoppelten Entscheidungen in die Turbo-Schleife die Detektion verbessern. Um mögliche Gewinne evaluieren zu können, wird in der Arbeit ein adaptiver Max-Log-MAP-Algorithmus für die Entzerrung entwickelt. Die Simulationsergebnisse zeigen, dass die Zuverlässigkeitsinformationen nach dem Entzerrer verbessert werden. Auch können die Zeitvarianzen nahezu kompensiert werden. Durch die erneute Kanalschätzung bei jeder Iteration ist der Iterationsgewinn immer noch nicht signifikant. Eine Verwendung der Turbo-Detektion für GMSK-basierte GSM-Dienste scheint somit nicht sinnvoll.

Bei Daten-Diensten, wie GPRS, wird zusätzlich zu dem Faltungscode noch ein CRC-Code verwendet. In den Untersuchungen wird der CRC-Code in die Turbo-Schleife miteinbezogen. Es zeigt sich, dass die iterative Decodierung dieses seriell verketteten Codes mit SISO-Modulen keine Verbesserung bringt. Der einzige Gewinn im Vergleich zu herkömmlichen Detektionsverfahren ist auf die Verwendung eines Soft-In-Decoders für den CRC-Code zurückzuführen. Dieser Gewinn beträgt aber lediglich 0.5 dB und eliminiert gleichzeitig die Fehlererkennungseigenschaften des Codes. Um dieselbe Restfehlerwahrscheinlichkeit wie mit herkömmlichen Decodern zu wahren, wird dieser Gewinn wieder aufgebraucht. Der Einsatz von SISO-Decodern für den CRC-Code ist somit nicht gerechtfertigt.

Für hochratige Datendienste soll in GSM-Systemen zukünftig eine 8-PSK-Modulation verwendet werden. Diese Modulation bedingt eine stärkere Interferenz. Somit können für diese Datendienste Iterationsgewinne erwartet werden.

Um dies zeigen zu können, werden zunächst die paketvermittelten Dienste EGPRS untersucht. Hier zeigen Simulationen für Codierschemata mit mittlerer Übertragungsrate, dass bei rauschbegrenzten Szenarien für typische Kanäle Gewinne von ca. 2 dB erzielbar sind. Bei interferenzbegrenzten Szenarien reduziert sich der Gewinn auf ca. 1 dB. Auch in kritischen Szenarien wie bei hohen Geschwindigkeiten zeigt sich, dass

Turbo-Detektion die Fehlerraten signifikant reduziert. Eine genaue Beschreibung des Nutzens der Turbo-Detektion für EGPRS ist möglich, wenn der Datendurchsatz der ARQ-Protokolle für alle Codierschemata von EGPRS evaluiert und anhand der Signalpegelverteilung in der gesamten Zelle bewertet wird. Die Untersuchungen zeigen, dass die spektrale Effizienz für rauschbegrenzte Szenarien um 10 bis 30% gesteigert werden kann. Bei interferenzbegrenzten Szenarien ist eine Steigerung von 5 bis 12% möglich.

Ein weiterer hochratiger Datendienst ist der erweiterte verbindungsbezogene Datendienst ECSD. Im Laufe der Standardisierung wurde zur Codierung ein seriell verketteter Faltungscode vorgeschlagen. Dieser verkettete Code zusammen mit dem Übertragungskanal kann als doppelt verketteter Faltungscode interpretiert werden. Die Einbeziehung des Entzerrers in die iterative Decodierung gewährleistet zusätzliche Verbesserungen zu der einfachen Turbo-Decodierung. Die Gewinne können um 1 bis 2 dB gesteigert werden.

1 Introduction

In the last decades mobile communications is one of the fastest growing markets within the area of telecommunications. The break through started at the beginning of the eighties when the mobile cellular networks of the “first generation” were introduced. All these systems, e.g. AMPS, TACS, NMT, C-450, are based on analog transmission. Right after the introduction of these systems it became obvious for several european countries that even these new analog systems have strong limitations. On the one hand, the systems were designed for only a small number of subscribers. The demand for mobile telephony was underestimated and it became clear that the new systems would come up against capacity limits soon. On the other hand, it was not possible to use the same mobile phone within several countries because of the non-compatible standards in various countries. Hence, the “groupe spéciale mobile” (GSM, later changed to “global system for mobile communications”) was already founded in 1982 inside the CEPT (conférence européene des postes et télécomminications). The goal of this group was to develop a new mobile radio system that avoids the deficiencies of the first generation systems and meets the demands of the fast growing market of mobile telephony by considering bandwidth efficient techniques and incorporating new services. In 1990 the GSM-standard phase 1 was settled and supplemented with phase 2 in 1995. In GSM, like in other “second generation” mobile radio standards e.g. D-AMPS, PDC, digital transmission techniques are used. Since the first commercial GSM-network started, the number of subscribers grew from one million in 1993 to 138 million in 1998. A deceleration of this growth is not foreseen. Today, several GSM-networks have already reached their limits, i.e. the allocated spectrum is not sufficient any more to support the growing demand for mobile communications. Additionally, the demand for more transmission capacity is accelerated because of the high rates of growth of data transmission in telecommunications, e.g. multimedia applications and internet. These new services require higher and more flexible data rates. In order to meet the requirements of these services, new standards are specified within GSM phase 2+. Packet and circuit switches services like GPRS (general packet switched services) and HSCSD (high-speed circuit switched data) are introduced. In a further step higher-order modulation techniques are specified to enhance the data services of GPRS and HSCSD, and to increase the spectrum efficiency. This new stage is called EDGE (enhanced data rates for GSM evolution). In this part of GSM phase 2+, 8-PSK (phase shift keying) modulation enable data rates up to three times higher than in conventional GSM-services, e.g. 69.2 kbit/s per time slot. Due to the higher-order modulation, the data transmission becomes more susceptible to signal distortions of the mobile radio environment. Hence, the spectrum efficiency cannot be tripled. In order to increase the capacities of GSM, modern techniques of information and coding theory are essential.

The birth of information theory is associated with the work of C. E. Shannon [Sha48] in 1948. In this and in later works, e.g. “Communications in the presence of noise”

[Sha49], C. E. Shannon proved that error free transmission is possible at a certain signal energy if the data is transmitted below the so-called Shannon limit. Unfortunately, Shannon could not show how efficient transmission and channel coding have to be designed in order to reach the Shannon limit.

Since C. E. Shannon also showed that transmission quality can be improved by increasing the word length, the focus of research was to develop powerful codes of large blocklenghts. The complexity of these coding schemes is very high, resulting in a non-implementable decoding complexity. A new coding strategy was to break up the long codes into several small, concatenated codes [For66a]. Using these codes, the decoding complexity does not grow exponentially with the length of the codes. Today, the fast progress in semiconductor industry allows the implementation of complex decoding algorithms. These algorithms enable that, apart from the transfer of hard-decisions, reliability information of the decisions is passed between the decoding modules [Hag94]. These modules are called SISO (soft-in/soft-out) modules. They distinguishably process the so-called “soft-values” and generate “soft-decisions” for the next decoding unit. The exploitation of the reliability of the decisions increases the efficiency of the decoder.

A further milestone of coding theory are the so-called “turbo-codes” which were introduced by Berrou, *et al.* in 1993 [BGT93]. Based on the principle of iterative decoding [BDG79, LYH93], iterative decoding of parallel concatenated convolutional codes was proposed. Using this method, the Shannon limit can be approached. In [HNB97] the authors showed how to come as close as 0.27 dB to the Shannon limit by means of parallel concatenated Hamming codes. Besides the examination of the properties of parallel concatenated convolutional codes and blockcodes [BeG96, BeM96, HOP96], also serially concatenated convolutional codes and blockcodes are investigated [BeM96c, BDM98b].

A main interest lies in the application of the turbo-principle to other detection schemes [Hag97]. This principle relies on the turbo-component that is fed back from decoding units to former decoding units in order to improve the decisions. One of the applications is iterative equalization and decoding, i.e. the so-called turbo-detection. Firstly published in [DJB95], turbo-detection was further examined in [BKH97, PDG97, GLL97]. In this method the channel encoder and the transmission channel are regarded as a serial concatenation of convolutional codes. This code system can be iteratively decoded. As turbo-detection requires modification of the transmission only at the receiver, it can be adopted to existing mobile radio systems without an amendment of the transmission standard. Therefore, turbo-detection constitutes a possibility to improve the performance of existing and future transmission standards.

In this work the performance of turbo-detection for existing GSM-services is examined. Additionally, new services using higher-order modulation techniques are investigated. This work demonstrates the benefits and limiting factors of turbo-detection for GSM-systems.

In Chapter 2 the fundamental concepts of mobile digital transmission are described. The principles are explained by the example of GSM, but are also valid for other digital TDMA-(time-division multiple access)-systems. After a short introduction to channel encoding and SISO channel decoding, the transmission channels are explained. Modulation as well as mobile radio channels are treated. Further, the equalization for TDMA-systems is described. Channel parameter estimation and symbol-by-symbol MAP (maximum-a-posteriori) equalization are discussed.

The principles of iterative equalization and decoding are explained in Chapter 3. The component decoders, i.e. the equalizer and the channel decoder, and the turbo-component in general are examined. Because block-diagonal interleaving is used in important GSM-services, e.g. TCH/FS (traffic channel at fullrate speech), turbo-detection cannot be applied on these services without modifications of the original turbo-detection scheme. In block-diagonal interleaving, the information of successive code blocks is interleaved together. Hence, this interleaving scheme is called interblock interleaving throughout this thesis. Interblock interleaving offers several possibilities to detect the data iteratively. Various processing schemes are introduced and are compared according to their complexity, delay and error performance. In order to be able to compare the iteration gains, first examinations are carried out for channels with severe frequency selectivity because the obtained iteration gains are large. Further, the influence of the channel parameter estimation and two various detection algorithms, i.e. the Log-MAP and the Max-Log-MAP algorithm, are investigated.

Chapter 4 shows the performance of turbo-detection for various modulation techniques of GSM-systems. In the first part GMSK-(Gaussian minimum shift-keying)-modulation is examined. The service of special interest is the TCH/FS service. The detection schemes of Chapter 3 are applied to GMSK-modulation and to specified mobile radio channels of GSM. Based on the gained knowledge, turbo-detection is then applied to GPRS. Here, blockcodes are used to detect erroneous transmitted blocks. The decoding of these blockcodes is included in the turbo-loop. As from SISO decoding of these codes the error detection properties are lost, a new decoding strategy is developed. This new decoding scheme exploits the soft-inputs and keeps the error detection properties. The performance of this scheme is then examined for GPRS transmission.

In the second part of Chapter 4 turbo-detection is applied to 8-PSK modulated signals. At the beginning, the necessary modification of the equalizer algorithms for higher-order modulation techniques are described. Then, turbo-detection is examined for the packet switched services of EDGE. Several propagation conditions are included in these examinations. Finally, turbo-detection is treated in combination with iterative decoding of serially concatenated convolutional codes. These codes are proposed for circuit switched services in EDGE. Simulation results illustrate the behaviour of this extended turbo-detection method.

The investigation of Chapter 3 and Chapter 4 are performed with non-adaptive equalization techniques. However, if the mobile radio channels are strongly time-variant, detection performance suffers because of non-adaptive channel estimation. In Chapter 5, first, these effects are introduced. Second, the principles of adaptive equalization are explained. It is shown how adaptive equalization can be incorporated into the turbo-loop in order to improve channel estimation. The performance of this new technique is demonstrated for the TCH/FS service.

Conclusions and an outlook for further studies are given in Chapter 6.

Parts of this work are published in [BaF98a], [BaF98b], [BaF98c], and [BaF99].

2 Basic concepts

In this chapter the basic concepts for the reliable transfer of data via channels distorted by ISI (intersymbol interference), inherent due to limited bandwidth[1] and multipath propagation [Pro95, Par92], are introduced. These principles are essential to understand the new concepts developed throughout this work. As the focus of this work is to apply new iterative detection principles to mobile communication systems, attention is paid on the detection of signals in time-variant environments. This work follows the concepts of the GSM transmission scheme [ETS98b, MoP92]. However, the model used in this thesis also comprises the fundamentals of other present TDMA mobile communication systems, e.g. PDC (pacific digital cellular) and USDC (United States digital cellular) also known as D-AMPS (digital advanced phone service) [Rap96,Per92].

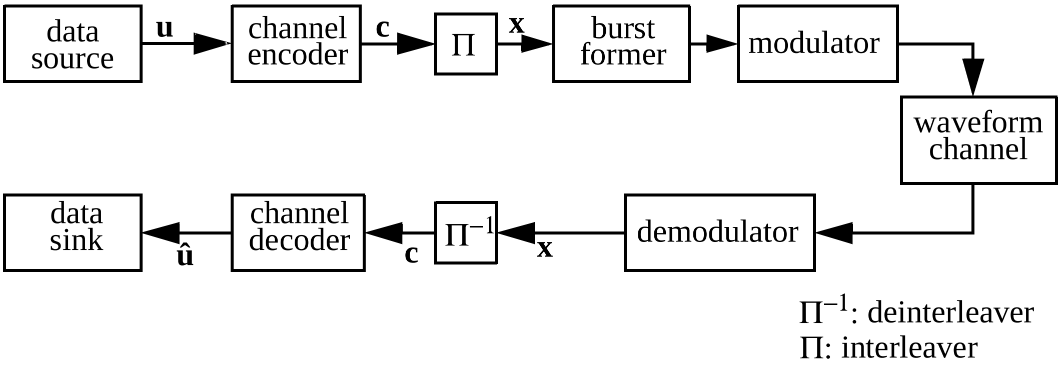


Figure 2.1. Generic communication system model

A generic communication system can be described by the block diagram shown in Fig. 2.1 [Pro95, ETS98b]. The data source generates information sequences **u**. These information sequences are grouped in blocks. They can arise from sources such as a speech encoder. The information sequences are first processed by a channel encoder where suitable error control coding is employed. Unlike the speech coder, where redundancy is removed, redundancy is added in order to improve transmission reliability. Two error control coding mechanisms are exploited. An outer coder is used for error detection and an inner code is used for error correction. Both principles are explained in Section 2.1. After the coded sequences **c** are interleaved (Π), a burst former multiplexes the sequences **x** to bursts, having a burst format comprising other data such as that used for channel estimation. Pilot, control, and guard information is added. Then, the modulator transforms the sequences into analog signals. These signals are send via the channel, i.e. the mobile radio channel. The mobile environment imposes strong distortions on the signals. These distortions are due to multipath propagation and motion of both

[1] In GSM, a Gaussian pulse with a bandwidth-time product of 0.3 is used. This pulse causes intersymbol interference.

the mobile user and other objects [Lee88]. At the receiver the signals from the antenna are amplified, filtered, converted, and sampled [Viz95]. Besides filtering, down conversion, and sampling, the demodulator detects the encoded signal and, by taking into account the constraints of the signal distortions, gives an estimate $\hat{\mathbf{x}}$ of the transmitted sequences **x**. These estimates are then deinterleaved and passed to the channel decoder. The channel decoder generates an estimate **û** about the information sequence **u** by considering the constraints of the code. These estimates are then passed to the data sink, e.g. a speech decoder.

Before starting a detailed description of the components of the transmission model, the so called soft-values, also called algebraic values [BDG79], have to be introduced. In the above description of the transmission model components only hard-decisions, i.e. estimates of the transmitted bits, are passed between the receiver. In order to fully exploit the information available at all stages of the receiver not only the estimate, $\hat{x}_i$, of the transmitted bit, x_i, but also the reliability of this estimate should be forwarded to the next component of the receiver [Hag92]. The estimate together with its reliability is called soft-value. A representation of a binary[1] soft-value is the log-likelihood ratio or algebraic value $L(\quad)$ also called L-value denoted by

$$L(\hat{\quad}) = \ln\frac{P(\qquad)}{P(\qquad)}. \tag{2.1}$$

The sign of the soft-value, $L(\quad)$, denotes the hard information, $\hat{x}_i$, and the magnitude is a measure of the reliability.

The advantage of the soft-values is obvious. Assuming a bursty channel, the demodulator receives unreliable information about the data transmitted during the deep fades. If the demodulator only passes the hard information, the decoder treats this decision with the same priority as the decisions coming from less faded signals. By taking into account the probabilities of these decisions, the decoder is able to give a more reliable estimate of the transmitted data.

From the decoder point of view the transmission components starting at the interleaver and ending at the output of the deinterleaver can be regarded as a channel. If the demodulator generates hard decisions, information is lost and the quality of the detection is reduced [Hag94].

[1] There also exist soft-values of non-binary data. For the moment, the explanation is restricted to the binary case for clarity. The concept of soft-values can be easily adapted to the non-binary case as will be done in Chapter 4.

2.1 Channel coding

The purpose of each coding system is to add redundancy efficiently to an information sequence in order to protect this information sequence by correction of erroneous reception [LiC83, Bos98, Fri95]. The goal is to enable the reliable transfer of information at an affordable encoding and decoding complexity. The processing delay and the signal power as well as the code rate have to be considered.

For channel codes two major classes can be distinguished:

- convolutional codes, and,
- block codes.

In contrast to block codes, convolutional codes need not be restricted to finite bit streams. With convolutional codes, a continuous stream of encoded symbols can be processed without blockwise processing. This difference is not important for this thesis as only the transmission of information blocks is considered. The convolutional codes treated in the following are terminated. In this case, the convolutional code is also a block code. Compared to other block codes, high coding gains can be achieved with low complexity coders and decoders that apply maximum-likelihood detection. Therefore, convolutional codes are used for error correction in many communication systems.

The block codes considered in this work are CRC(cyclic redundancy check) codes. The CRC codes that are in general used for error detection belong to the class of cyclic block codes. Their hard-decision error detection capabilities can be fully exploited with low decoder complexity. Error detection is mainly applied in two variants. On one hand, if data such as speech data is transmitted, error detection can be used for error concealment [VaF98]. Once the error detection unit recognizes that the transmitted block is erroneous, the block is not forwarded to the speech decoder. The error can be concealed by methods such as passing the last correct block to the speech decoder. On the other hand, error detection can be combined with an ARQ (automatic repeat request) scheme. ARQ transmission is mainly used for non-realtime data streams. On erroneous detection, the data block is refused and retransmission is requested until the block is correctly detected.

In Fig. 2.2, a block diagram for a typical concatenated channel coder for error correction and detection is depicted. First, the block coder adds redundancy for error detection. Second, the convolutional coder adds redundancy for error correction.

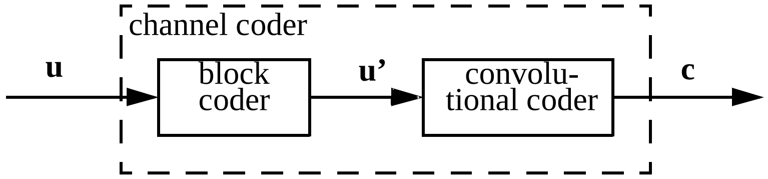


Figure 2.2. Concatenated coding scheme for error detection and error correction

In Fig. 2.3, the processing of the data during encoding of an information block **u** with a

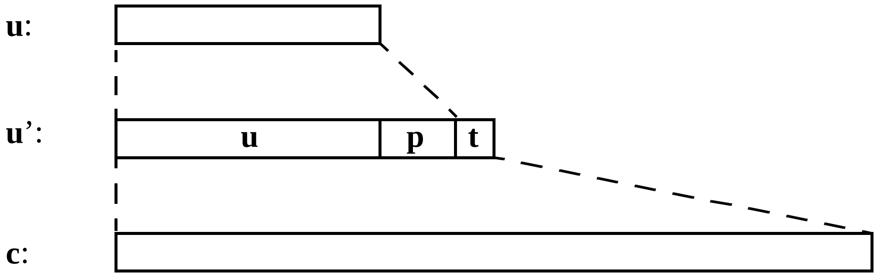


Figure 2.3. Encoding of the information sequence **u**

systematically encoded[1] block code is depicted. The systematic encoder appends a parity sequence **p** to the information sequence **u**. Then, tail bits **t** are appended for trellis termination of the following encoder, as will be explained later. The intermediate coded sequence **u**' is again encoded by a convolutional encoder and the coded sequence **c** is obtained.

Despite the differences of block codes and convolutional codes, they can be formally described in a similar manner. In the following sections the properties of these codes are introduced using a uniform description.

2.1.1 Linear block codes

An information sequence **u** of length k, denoted by

$$\mathbf{u} = (u_0, u_1, \ldots, u_{k-1}), \tag{2.2}$$

is transformed by the channel encoder to the encoded sequence **c** of length n with

[1] The systematically encoded code will be introduced in the next chapter.

$$\mathbf{c} = (c_0, c_1, \ldots, c_{n-1}) . \tag{2.3}$$

The symbols u_i and c_i are elements of the Galois field 2: $u_i, c_i \in F = GF(2)$. Therefore, the code vectors **c** are in the n-dimensional vector space F^n; the information vectors, in the k-dimensional vector space F^k.

Note that, for the linear block code *C* the linearity property is fulfilled:

$$\mathbf{c}_i, \mathbf{c}_j \in C \Rightarrow \mathbf{c}_i + \mathbf{c}_j \in C . \tag{2.4}$$

This means that the sum of two code words has to be a valid code word again.

The linear (n,k)-block code *C* can be described by the $(k \times n)$ generator matrix **G,**

$$C = \{\mathbf{uG} | \mathbf{u} \in F^k\} , \tag{2.5}$$

or via its $(n - k \times n)$ parity check matrix **H,**

$$C = \{\mathbf{c} \in F^n | \mathbf{cH}^T = 0\} \quad \text{with} \quad \mathbf{GH}^T = 0 . \tag{2.6}$$

Note that $\mathbf{G} \in F^{k, n}$ and $\mathbf{H} \in F^{n-k, n}$. $F^{k,n}$ is the $(k \times n)$- dimensional matrix of the field *F*.

A code whose code words are divided into an information part and a redundant checking part is encoded systematically. The generator matrix **G** has the following form **G**=(**P | I**$_k$) with **P** being the parity submatrix and $\mathbf{I}_k$ being the $(k \times k)$ identity matrix.

A code word, **c**, is generated by $\mathbf{c} = \mathbf{uG}$. At the decoder, the received sequence $\mathbf{r} \in F^n$ can be described by the superposition of the code word **c** and an error vector $\mathbf{e} \in F^n$. The decoder calculates the syndrome $\mathbf{s} \in F^{n-k}$ by

$$\mathbf{s} = \mathbf{rH}^T = (\mathbf{c} + \mathbf{e})\mathbf{H}^T = \mathbf{eH}^T . \tag{2.7}$$

If the error sequence **e** is not a valid code word, the syndrome **s** is not equal to the zero-sequence. Therefore, an error can be detected.

In order to determine the error correction and detection capability of a code an important parameter is the minimum distance d_{min} of the code. The Hamming weight w(**c**) is defined as the number of non-zero components of **c**, while the Hamming distance $d_H(\mathbf{c}_i,\mathbf{c}_j)$ of two code words $\mathbf{c}_i$ and $\mathbf{c}_j$ is defined as the number of positions the two words differ. The minimum distance d_{min} denotes the minimum Hamming distance of two code words. For a linear code the minimum distance is equal the minimum Hamming weight of a code word:

$$d_{\min} = \min\{ \quad \_\_ \quad | \qquad \}. \tag{2.8}$$

Each error vector **e** of weight $w(\mathbf{e}) < d_{\min}$ cannot change the code word into another code word. On error detection the received word can be detected as erroneous. Note that also a large fraction of error vectors **e** of weight $w(\mathbf{e}) \geq d_{\min}$ can be detected, as these error vectors need not necessarily be code words.

For error correction the decoder must provide an estimate **û** about the information sequence **u**. As there is a one-to-one correspondence between **u** and **c**, an optimum decoder has to minimize the decoding error probability $P(E) = P(\mathbf{c} \neq \mathbf{c}) = P(\mathbf{c} \neq \mathbf{c}|\mathbf{r})P(\mathbf{r})$, where $P(\mathbf{c} \neq \mathbf{c}|\mathbf{r})$ is the conditional error probability. As P(**r**) is assumed to be identical for all code words, P(E) can be minimized by minimizing $P(\mathbf{c} \neq \mathbf{c}|\mathbf{r})$ or by maximizing the conditional APP (a posteriori probability) $P(\mathbf{c} = \mathbf{c}|\mathbf{r})$. A decoder that maximizes $P(\mathbf{c} = \mathbf{c}|\mathbf{r})$ is called a MAP-sequence-decoder. Applying the Bayes rule, this probability can also be written as

$$P(\hat{\mathbf{c}} = \mathbf{c}|\mathbf{r}) = \frac{P(\mathbf{r}|\mathbf{c} = \mathbf{c})P(\mathbf{c})}{P(\mathbf{r})}. \tag{2.9}$$

If all information sequences, and hence all code words, are equally likely, then maximizing $P(\mathbf{c} = \mathbf{c}|\mathbf{r})$ is the same as maximizing the likelihood $P(\mathbf{r}|\mathbf{c} = \mathbf{c})$. A decoder that maximizes $P(\mathbf{r}|\mathbf{c} = \mathbf{c})$ is called a ML(maximum likelihood)-decoder. Note that the MAP-sequence-decoder and the ML-decoder are identical if no a priori information is available.

In addition to the MAP-sequence-decoder, the symbol-by-symbol MAP-decoder maximizes the probability $P(c_i = c_i|\mathbf{r})$ of each symbol c_i.

On a BSC (binary symmetric channel) the ML-decoder assigns the received sequence **r** to the code word **c** with the minimum distance $d_H(\mathbf{c},\mathbf{r})$. If the error weight $w_h(\mathbf{e})$ meets the condition $w_h(\mathbf{e}) \leq \lfloor (d_{\min} - 1)/2 \rfloor$ then **r** is assigned to the correct code word.

The higher the minimum distance $d_{\min}$ the better are the error detection and correction capabilities of a code.

2.1.1.1 Cyclic block codes

Cyclic block codes were introduced by E. Prange [Pra57]. The main advantage is the low encoding and decoding complexity. When encoding regular linear block codes the input sequence has to be multiplied with a $(k \times n)$-matrix. For syndrome decoding the received sequence has to be multiplied with a $(n - k \times n)$-matrix. As these matrices have to be stored the storage requirements become large for high values of k and n. For cyclic codes, only the generator polynomial g(D) and the check polynomial h(D), as will be defined in the following, are needed.

A linear block code (n,k) is called cyclic if the cyclic shift of a code word is also a code word:

$$(c_o,c_1, \dots, c_{n-2}, c_{n-1}) \in C \Rightarrow (c_{n-1},c_0, \dots, c_{n-3}, c_{n-2}) \in C\,. \tag{2.10}$$

For cyclic codes it is convenient to represent the sequences **u** and **c** with polynomials. The descriptions of (2.2) and (2.3) can be equivalently written in the following form [Bos98]:

$$u(D) = u_o + u_1D + u_2D^2 + \dots + u_{k-1}D^{k-1} \text{ and} \tag{2.11}$$

$$c(D) = c_o + c_1D + u_2D^2 + \dots + c_{n-1}D^{n-1}\,. \tag{2.12}$$

Cyclic codes can be described using the generator polynomial g(D) instead of the generator matrix **G** and the parity check polynomial h(D) instead of the parity check matrix **H**. The polynomials g(D) and h(D) are factors of D^n-1.

Now, equations (2.5) and (2.6) can be written as

$$C = \{u(D)g(D)|u(x) \in F[D]^{k-1}\} \text{ and} \tag{2.13}$$

$$C = \{c(D) \in F[D]^{n-1}\big|c(D)h(D) = 0 \text{ in } F_n[D]\} \text{ with} \tag{2.14}$$

$$g(D)h(D) = 0 \text{ in } F_n[D]\,. \tag{2.15}$$

Code words can be generated by multiplying the information sequence u(D) with the generator polynomial g(D), where g(D) devides D^n-1. The problem in designing a special code is to find a pair g(D) and h(D) of certain degree. Since the degree of g(D) and the degree of h(D) have to sum up to n and as both as the factor of both polynomials have to be equal to D^n+1, it is often very difficult to find a cyclic code with suitable primitive length, i.e. the length n that fulfils the two above properties.

2.1.1.2 Shortened cyclic codes

If the code of suitable primitive length cannot be found, very often shortening is used. Given an (n,k) cyclic code *C*, the set of code words for which the q leading high-order information digits are identical to zero are considered. These 2^{k-q} code words form a linear subcode of **G**. If these q zero information bits are deleted from each of these code words, a set of 2^{k-q} vectors of length n-q is obtained resulting in the shortened (n-q,k-q) blockcode. The code is called a shortened cyclic code; however, it is not cyclic any more. This code has at least the same error-correcting capability as the code its derived from. The encoding can be accomplished in the same way. Only decoding has to be slightly modified [LiC83].

The encoding is defined by the generator matrix $\mathbf{G}' \in F^{k- \ , n-q}$. This matrix is generated by deleting q columns and q rows from the generator matrix **G** of the cyclic code. The parity check matrix $\mathbf{H}' \in F^{n- \ , n-q}$ is derived similarly from the check matrix **H** of the cyclic code.

All linear block codes can be systematicaly encoded. The systematic form has the same random error decoding properties as the non-systematic encoding, and additionally possesses the desirable property that the parity and the information part are separate.

Example 2.1 *In* [ETS96] *for GPRS coding scheme CS-2 to CS-4 a systematically encoded, shortened cyclic code is used. The generator polynomial g(D) of the 16 bit CRC (cyclic redundancy check) code is given by* $g(\ ) = \qquad = (\qquad)p(\ )$. *The polynomial p(D) is a primitive polynomial of degree 15. Therefore, the natural length of the code is* $n = \qquad = 32767$. *The block lengths specified for GPRS range from k=274 to k=440. The cyclic code is shortened.*

2.1.2 Convolutional codes

Convolutional codes were first proposed by [Eli54]. With convolutional codes coding gains can be achieved with relatively low complexity, whereas with block codes the same coding gains can be only achieved with very complex encoding/decoding schemes[1]. Therefore, in most communication systems convolutional codes are used for error correction [AnM91].

In this work only rate $R=1/n_0$ convolutional codes are considered, where n_0 is the number of code bits per input bit. In this thesis, attention is only paid to block transmission. In case in which only blocks are transmitted, formally block codes and convolutional codes are identical.

A convolutional code can be described in its algebraic form by n_0 generator polynomials $g^{(1)}(D),\ldots,\ g^{(\ )}(\ )$. The generator polynomials are of order m, where m is the memory length of the code. Therefore the constraint length of the code is m+1. The information sequence u(D) is convolved with the generator polynomials $g^{(i)}(D)$ to form the sequence $c^{(i)}(D)$. The sequences $c^{(1)}(D)$, $c^{(2)}(D)$, ..., $c^{(\ )}(\ )$ are then multiplexed to the code sequence c(D). The generator polynomials can be combined in the generator matrix $G(D)=(g^{(1)}(D),\ldots,\ g^{(\ )}(\ ))$, where G(D) is a $(1 \times n_0)$ dimensional matrix

[1] This property is valid for codes of low code rate.

with elements from F[D]. The relation between the input and the output sequence can be written as

$$c(D) = u(D)G(D). \tag{2.16}$$

A non-recursive encoded convolutional encoder can be specified by a non-recursive shift register (see Fig. 2.4). In this figure a rate R=1/2 convolutional encoder of memory length m is illustrated.

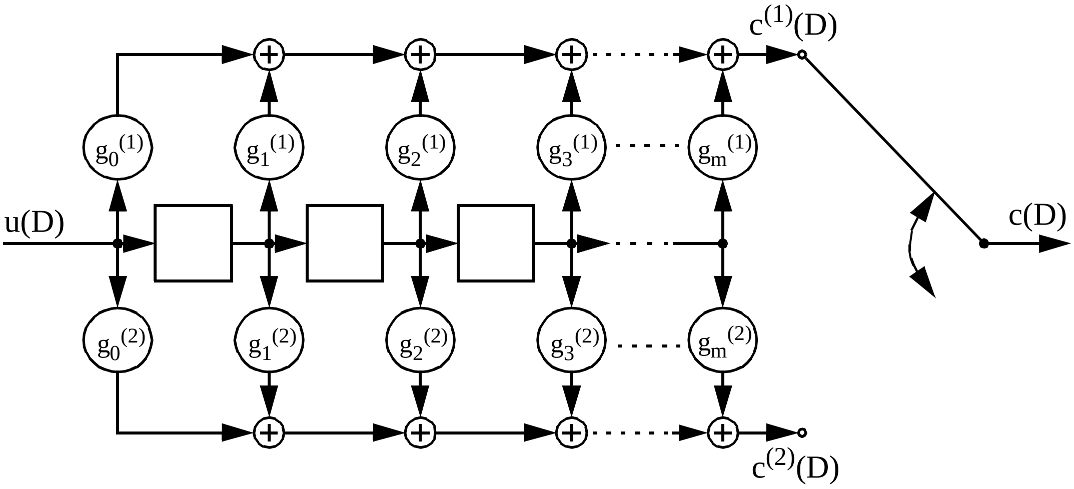


Figure 2.4. Non recursive convolutional encoder

Example 2.2 *In* [ETS98b] *the following generator polynomials for the rate R=1/2 convolutional coder for TCH/FS are specified:* $g_0(D) = 1 + D^3 + D^4$ *and* $g_1(D) = 1 + D + D^3 + D^4$. *The memory m is equal to 4.*

Another important class of convolutional encoder are recursively encoded convolutional codes. The output of the shift register is fed back to the input. These codes are used, for example, for concatenated convolutional codes. In Fig. 2.5 the shift register for a rate R=1 recursively encoded convolutional code is depicted. The generator matrix G(D) has the following form: $G(\quad) = (\qquad\qquad)$.

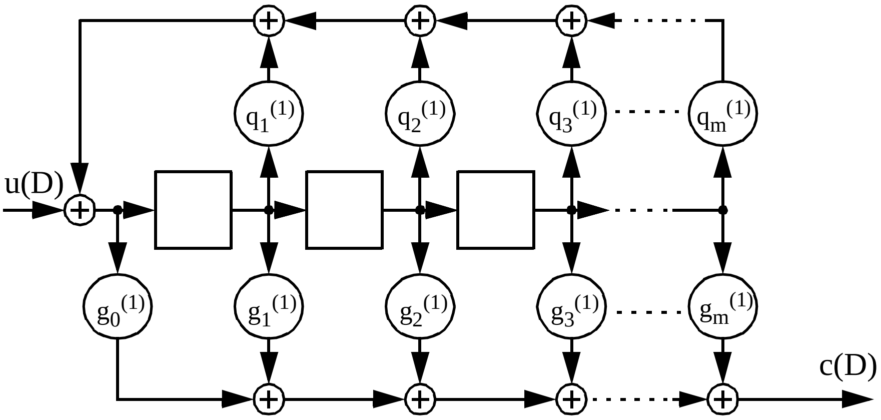


Figure 2.5. Recursive convolutional encoder

Example 2.3 *In* [ETS98f] *for ECSD (enhanced circuit switched data) a SCCC (serially concatenated convolutional code) is proposed. The rate R=1 inner convolutional code is recursively encoded with the generator polynomial* $g(D) = 1/(1+D)$.

A possible representation of the encoder is a trellis diagram. A trellis is a delineation of the state transitions of a finite-state machine over time. The trellis representation will be used throughout this work for the description of equalization and decoding techniques. In Fig. 2.6 a) and Fig. 2.6 b) the trellis diagram of a non-recursive and a recursive convolutional code of rate R=1/2 and memory m=2 are depicted. Depending on the current state S_i and on the current information bit u_t the code bits $c_i^{(1)}$ and $c_i^{(2)}$ are generated. If the current information bit u_t is a zero, the corresponding transition is called 0-transition; else, 1-transition. From the trellis diagram it is seen that each state has two possible successor states depending on the kind of transition. On non-recursive convolutional encoding, two transitions of the same kind merge; on recursive convolutional encoding, one 0-transition and one 1-transition merge.

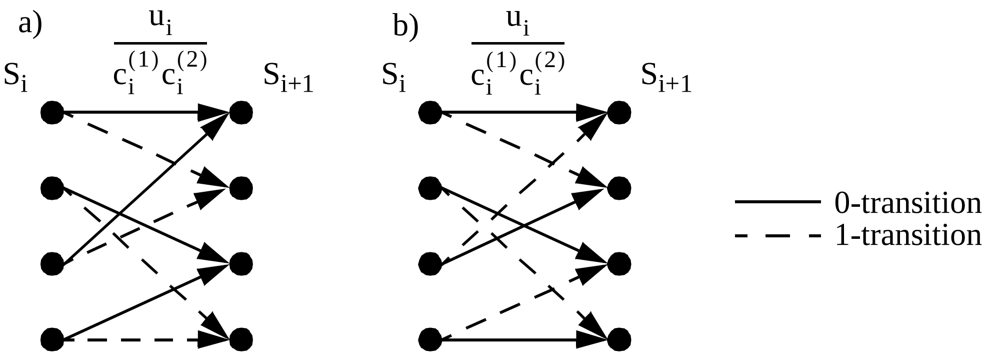


Figure 2.6. Trellis diagrams for non-recursive and recursive convolutional coders of rate R=1/2 and memory m=2.

2.1.3 Interleaving

In GSM-systems the transmission channel is not memory-less for two reasons: first, the used modulator and the delayed echoes at the receiver introduce ISI [Ste92]; second, due to multipath transmission the radio channel suffers from correlated fading [Lee82]. Detecting signals distorted by ISI and by correlated fading causes correlated errors at the output of the demodulator [Pro95]. Hence, the channel comprising the transmission channel and the demodulator has memory.

Convolutional codes are designed to combat randomly distributed, statistically independent errors [AnM91] which may occur in memory less channels. In order to achieve robust error correction on channels having memory, interleaving combined with convolutional coding constitutes an appropriate means. Prior to modulation and transmission, the coded sequence is interleaved. The interleaver maps the coded sequence one-to-one to the output sequence by rearranging the order of the symbols.

At the receiver the symbols are demodulated. The symbol estimates are correlated [Pro95]. The deinterleaver, carrying out the reverse process of the interleaver, decorrelates the relative positions of the symbols respectively in the demodulator output and in the decoder input. Error bursts are rearranged to single errors or at least to smaller error bursts. Due to interleaving the decoder can decode the more statistically independent data [LiC83].

The interleaver/deinterleaver scheme combats the statistical dependency due to ISI and, together with the decoder, exploits the time-variance of the channel.

Although interleaving is an effective way of increasing error correction capabilities, its main disadvantage is the introduction of delay [Ste92]. A large interleaver causes long delays. Hence, a compromise between delay and error performance must be found. Particularly on real time data transmission a certain delay must not be exceeded.

To improve the interleaving gain without introducing high delay, interblock-interleaving can be used [Ste92]. Here, the coded block is interleaved together with other blocks.

Example 2.4 *In the following a look is taken at the GSM block diagonal interleaver as specified for the TCH/FS in* [ETS98b]. *As shown in Fig. 2.7 every 20 ms a speech frame is encoded and interleaved on eight bursts. At the time instant t_0 speech frame 1 is available from the speech encoder and has to be prepared for transmission. The frame is split and reordered into eight subblocks. Four subblocks together with four subblocks of speech frame 0 are mapped onto four bursts. For the transmission of these four bursts about 19 ms are needed. Only 1 ms later, at time instant t_1, the data of the next speech block is available. The remaining four subblocks of the current frame 1 together with four subblocks of the next frame 2 are now mapped onto the next four bursts and transmitted. At time instant t_2 the current speech frame 1 has been completely transmitted. Transmitting one speech frame takes at least four bursts causing a delay of about 20 ms. As an interleaving depth of eight bursts instead of four bursts is used, an additional interleaving delay of 20 ms is introduced. If the data of block 1 and 2 were to be mapped on the same eight bursts, an additional delay of 20 ms would be imposed without enhancing interleaving gain.*

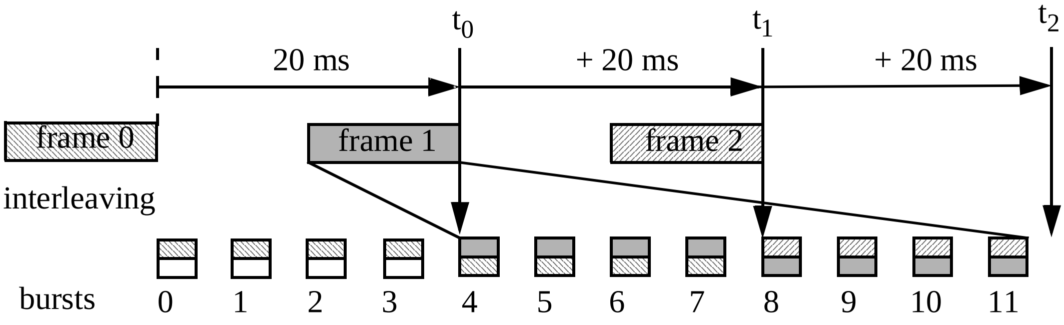


Figure 2.7. Block diagonal interleaving for GSM TCH/FS.

In the design of a transmission scheme the interleaver has to be matched to the coder in order to ensure good coding gains [BCT98]. Particularly when transmitting data over block fading channels, the interleaver must be carefully designed. The interleaving scheme has to be optimized so that the minimum block-Hamming distance, i.e. the minimum number of blocks in which two code words differ, is maximized [BCT98].

2.2 Soft-in/soft-out decoding

2.2.1 Soft-in/soft-out decoding for convolutional codes

As already mentioned, to fully exploit the information available at the receiver soft-values have to be passed between two consecutive receiver components. Therefore, it is necessary to use SISO devices on all stages of the receiver [Hag92]. This is also valid for the decoder.

The SISO decoder gets a priori values L(**c**) about the coded sequence **c** and generates a posteriori information $L(\mathbf{u})$ about the information sequence **u** [HaA96]. If additional knowledge L(**u**) about the information sequence is available, the SISO decoder should be also able to use this information. In [HiR98, HiR97, Hag95] it is shown that using a priori information can improve decoding reliability.

The block diagram of a SISO decoder is depicted in Fig. 2.8.

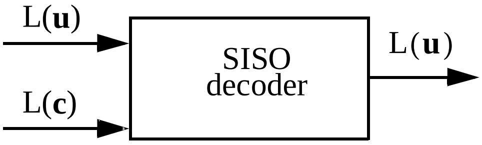


Figure 2.8. SISO decoder

The decoder has to deliver the APP $P(u_i = 0|L(\mathbf{c}), L(\mathbf{u}))$, the probability that the bit u_i is equal to zero under the condition that the input sequences L(**c**) and L(**u**) are observed [BCJ74]. Equivalently its log-likelihood value $L(u_i)$ about each bit i of the sequence **u** can be given:

$$L(\hat{u}_i) = \ln\frac{P(u_i = 0|L(\mathbf{c}), L(\mathbf{u}))}{P(\hat{u}_i = 1|L(\mathbf{c}), L(\mathbf{u}))}. \tag{2.17}$$

From the L-values $L(\mathbf{u})$ the hard-decisions $\mathbf{u}$ can be derived by considering their signs:

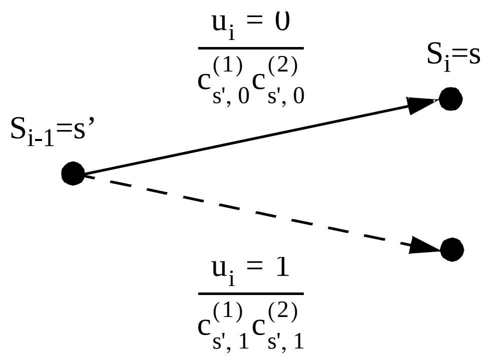
$$\hat{u}_i = \begin{cases} 0, \text{ if } L(u_i) \geq 0 \\ 1, \text{ if } L(\hat{u}_i) < 0 \end{cases}. \tag{2.18}$$

The decoding process can be described by the use of the trellis diagram. From the generator polynomials of the convolutional code the decoder can calculate the bits

$c_{s,0}^{(\ )}, \quad , c_{s,0}^{(\ \ )}$ and $c_{s,1}^{(\ )}, \quad , c_{s,1}^{(\ \ )}$ corresponding, respectively, to the 0-transition or the 1-transition from state s[1]:

$$c_{s,0}^{(\ )} = \sum_{l=1}^{\cdots} g_l^{(\ )} u_{k-l}, \tag{2.19}$$

$$c_{s,1}^{(\ )} = \qquad = c_{s,0}^{(\ )} + 1\,. \tag{2.20}$$



$u_i = 0$
$c_{s',0}^{(1)} c_{s',0}^{(2)}$
S_i=s
S_{i-1}=s’
$u_i = 1$
$c_{s',1}^{(1)} c_{s',1}^{(2)}$

Figure 2.9. 0- and 1-transition with the corresponding code bits originating at state s’.

For all trellis-based decoder algorithms it is essential to calculate the transition probabilities $\gamma_i(s', s) = P(S_i = s, L(c_i), L(u_i) | S_{i-1} = s')$ or their natural logarithms $\Gamma_i(s', s) = \ln\{\gamma_k(s', s)\}$, [RHV97].

For a certain stage k in the trellis, the information $L(u_k)$ and the information $L(c_i^{(\mu)})$ with $\mu \in \{1, \ldots, n_0\}$ are known. The normalized transition probabilities or, equivalently, its logarithms can now be calculated. $\Gamma_{i,0}$(s’,s) corresponds to a 0-transition and $\Gamma_{i,1}$(s’,s) to a 1-transition:

$$\Gamma_{i,0}(s', s) = \log\{\gamma_{i,0}(s', s)\} = L(\hat{u}_i) + \sum_{\mu=1}^{n_0} (1 - c_{s',0}^{(\mu)}) L(\hat{c}_i^{(\mu)}), \tag{2.21}$$

$$\Gamma_{i,1}(s', s) = \log\{\gamma_{i,1}(s', s)\} = \sum_{\mu=1}^{n_0} c_{s',0}^{(\mu)} L(\hat{c}_i^{(\mu)})\,. \tag{2.22}$$

Equations (2.21) and (2.22) are valid because $c_{s',0}^{(\mu)} = (1 - c_{s',1}^{(\mu)})$, see (2.20).

The various SISO decoding algorithms only differ in the way these normalized probabilities are processed. An optimum algorithm that calculates the APP is the symbol-by-

[1] It is assumed that for all codes $g_0^{(i)}$ is equal to one. This assumption is valid for all codes used throughout this thesis. It can be also stated that this assumption holds for most cases as the ODP (optimum distance profile) convolutional codes have this property [JoZ99].

symbol MAP algorithm, also called BCJR algorithm, introduced in [BCJ74]. In this work this algorithm is referred to as the MAP algorithm. The APP are calculated by the following equation:

$$L(\hat{u}_i) = \ln \frac{\sum\limits_{(s', s), u = 0} P(S_i = s', S_{i+1} = s, L(\mathbf{c}), L(\mathbf{u}))}{\sum\limits_{(s', s), u = 1} P(S_i = s', S_{i+1} = s, L(\mathbf{c}), L(\mathbf{u}))} . \qquad (2.23)$$

To prevent numerical problems and also to reduce computational complexity, the calculations of the MAP algorithm can be executed in the log-domain [RHV95]. The algorithm is then called the Log-MAP algorithm. The Log-MAP algorithm is still optimum. A further decrease in complexity can be achieved by using suboptimum derivatives, e.g. Max-Log-MAP algorithm and SOVA (soft-output Viterbi algorithm) [HaH89a]. The accuracy of soft-output decreases [RHV97]; however, if non-iterative detection schemes are used, the performance of the system degrades only slightly [Fra96]. Iterative schemes are more sensitive to suboptimum soft-output calculation [RHV97]. Especially for other suboptimum variants, e.g. the M-algorithm, the performance decrease is high [FrA97, FrA98]. Hence, only the Log-MAP and the Max-Log-Map algorithm are treated in this thesis.

2.2.2 Soft-in/soft-out decoding of block codes

For block codes several algebraic soft-in[1] algorithms exist, e.g. the Chase-algorithm [Cha72], the GMD-algorithm [For66b], and the MLD-algorithm [KNI94]. These algorithms are not suited for soft-output decoding and hence are not treated in this thesis. These algorithms, as well as similar algorithms [FoL95, BeS86], are not suited for accurate soft-output generation. A new algorithm constructed for SISO decoding is the IDA-algorithm [LBB98]; however, this algorithm is suboptimum.

Similar to the SISO decoding of convolutional codes, block codes can also be decoded using the syndrome trellis. The syndrome trellis is minimal [Bos98]. Thus it has the lowest possible decoding complexity. The maximal number of states at any trellis depth is min(2^{n-k},2^k) for a (n,k)-binary block code [Wol78, BCJ74, Bos98]; only the construction of the trellis is different. Once the syndrome trellis is constructed, the detection follows the same rules.

The difference to convolutional codes is that, except for cyclic codes, the trellis of a block code is not periodic. The trellis has to be constructed for each stage anew. The

[1] Soft-in algorithms for block codes are normally referred to as soft-decision algorithms. In this work the term soft-in algorithms is used for clarity.

states of the trellis are calculated using the parity matrix **H**. With $\mathbf{h}^{(i)}$ being the ith column of **H**, the successor states S_i can be calculated by

$$S_i = S_{i-1} + \mathbf{c}_i\mathbf{h}^{(\ )}. \tag{2.24}$$

The calculation of the transition probabilities has to be modified because the code bits $c_{s,\ ,0}$ and $c_{s,\ ,1}$ also depend on the position i. Instead of the generator, polynomial the ith column of generator matrix **G** is used for the calculation of the code bits corresponding to the transitions. Equations (2.19) and (2.20) are changed to

$$c_{s,\ ,} \quad = \sum_{l=1}^{\cdots} g_l^{(\ )} u_{i-l} \text{ and} \tag{2.25}$$

$$c_{s,\ ,} \quad = \qquad\qquad = c_{s,\ ,} \quad + 1, \tag{2.26}$$

with $\mathbf{g}^{(i)}$ being the ith column of the generator matrix **G**.

The logarithms of the transition probabilities are now calculated by:

$$\Gamma_{i,0}(\qquad) = \qquad\qquad = L(\quad) + (\qquad)L(\quad), \tag{2.27}$$

$$\Gamma_{i,1}(\qquad) = \qquad\qquad = c_{s',\ ,} \quad L(\quad). \tag{2.28}$$

For a (n,k)-binary block code the trellis has 2^{n-k} states. Now, the soft-output can be calculated according to (2.23).

For high-rate codes $(R > 1/2)$ a further reduction in computational complexity can be achieved by decoding with the dual code [HaR76, BDG79, HOP96]. Even though the amount of computations is reduced, the trellis representation would result in 2^k states. Note that the concept of dual codes can also be applied to convolutional codes [Rie98].

The maximal number of states at any trellis depth is $2^{\min(n-k,k)}$.

2.3 The transmission channel

The transmission channel model comprises several parts of the communication as depicted in Fig. 2.10. It reaches from the modulator to the output of the sampling device in front of the equalizer. In order to transmit digital data via a radio channel, the modulator maps this data to an analog waveform. From the transmit antenna the waveform is transmitted across the mobile radio channel to the receiver where it is detected at the antenna. Due to thermal noise of the receive amplifiers, AWGN (additive white Gaussian noise) is added [Kam92]. The noise of the first amplifier of the receiver front end has a larger impact on the signal than the others [Viz95] and, therefore, is placed in front of the down conversion unit. Here, down conversion is modelled simply by a single factor. In real systems, this downconversion is implemented in several steps [Viz95]. The receive lowpass filter h_{RX} removes the spectra from down conversion and suppresses ACI (adjacent channel interference). The output is sampled at symbol period T. This sampled sequence is then fed into the equalizer where the ISI is mitigated.

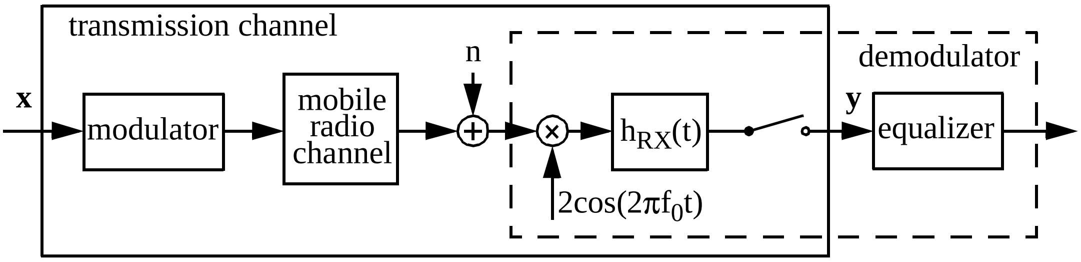


Figure 2.10. Components of the transmission channel

To get a better understanding of the transmission channel a closer look at its components is necessary.

2.3.1 Modulation

In this thesis only digital modulation techniques are treated. Analog modulation techniques have been used for mobile communication systems of the first generation, e.g. the C-Netz [Rap96], and are not of interest in this work.

As previously stated the modulator has to map the digital data onto an analog signal in order to transmit it via the analog channel. The modulated waveform of a digital modulation scheme can be described by

$$s(t) = \mathrm{Re}\{c(t)e^{j2\pi f_0 t}\} \tag{2.29}$$

with c(t) being the equivalent lowpass signal or the complex envelope of the signal and f_0 being the carrier frequency [Pro95].

There exist two different classes of digital modulation techniques: linear and non-linear modulation. Depending on the type of digital modulation, the lowpass signal can be expressed in various forms. For linear modulation the lowpass signal c(t) is described by the superposition of the responses of the transmit filter $h_{TX}(t)$ to the various symbols d_n:

$$c(t) = \sum_{n=-\infty} d_n h_{TX}(t - nT), \tag{2.30}$$

with $h_{TX}(t)$ being the impulse response of the impulse shaping transmit filter; and d_n, the input symbols. The incoming sequence **x** is split into blocks of b bits. Then, each block n of b bits is mapped onto a complex symbol $d_n=d_{In}+jd_{Qn}$. The real part d_{In} is called the inphase component; and the imaginary part d_{Qn}, the quadrature component. If the symbol space is M-ary, b is equal to ld(M). The impulse filter $h_{TX}(t)$ is excited by $d_n\delta(\qquad)$.

In Fig. 2.11 a linear modulator in its lowpass description is depicted. The real-valued data sequence **x** is split into i blocks $\mathbf{x}^{(1)}$,..., $\mathbf{x}^{(i)}$ and these blocks are converted to the complex-valued analog signal c(t).

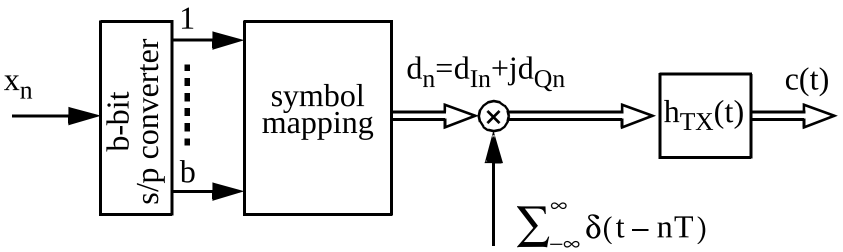


Figure 2.11. Linear modulator in lowpass representation

Depending on the type of symbol mapping, the linear modulation techniques can be distinguished, e.g. QAM (quadrature amplitude modulation), ASK (amplitude shift keying), PSK. There also exist derivatives of these types such as offset or differential PSK.

Example 2.5 *In EDGE the ETSI (european telecommunications standards institute) has agreed on a 8-PSK modulation scheme with a linearized GMSK (Gaussian minimum shift-keying) pulse shaping filter* [ETS98e]. *The symbol mapping of this modulation is illustrated in Fig. 2.12. According to the Gray code enumeration, three bits are mapped onto a symbol such that the 3-tuples of neighbouring symbols differ in only one bit position.*

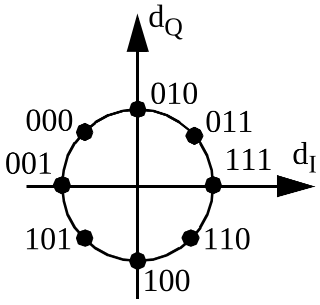


Figure 2.12. Gray code enumeration for 8-PSK.

The impulse response $h_{TX}(t)$ is depicted in Fig. 2.13. Strong ISI is imposed to the signal.

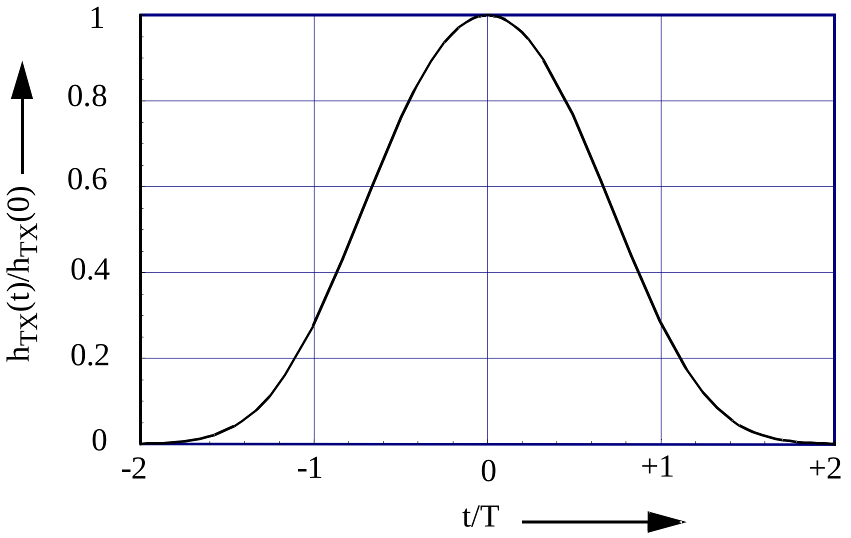


Figure 2.13. Pulse shaping filter of linearized GMSK.

Besides the class of linear digital modulation techniques there also exist non-linear digital modulation techniques. The signal cannot be expressed by the linear superposition of the responses from the different symbols. In general, a non-linear digital modulated lowpass signal c(t) can be described by

$$c(t) = \exp\left\{ j\pi h \sum_{n=-\infty}^{\infty} x_n \int_{-\infty}^{t-nT} h_f(\xi) d\xi \right\}, \tag{2.31}$$

with h being the modulation index and $h_f(t)$ the frequency pulse response [Pro95].

Example 2.6 *As a non-linear modulation technique GMSK is used in GSM* [ETS97]. *Its equivalent lowpass signal can be represented by*

$$c(t) = \exp\left\{ j\pi \sum_{n=-\infty}^{\infty} a_n \Phi(t-nT) \right\}, \tag{2.32}$$

where Φ(t) *is the phase pulse and a_n (*$a_n \in \{ \quad \}$*) the output of the differential encoder. The signal is generated as shown in Fig. 2.14. The relation between* Φ(t) *and the filter impulse responses h(t), g(t) is given in* [AAS86].

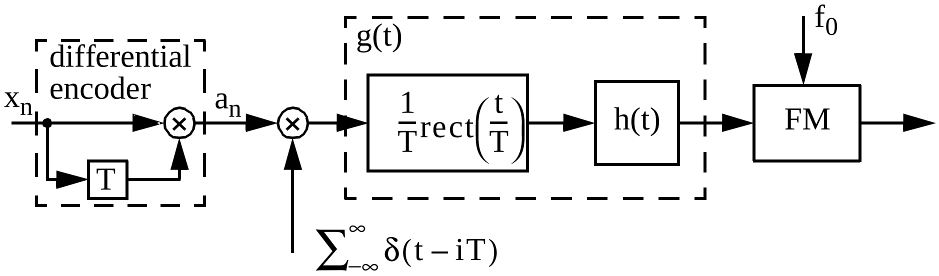

differential
encoder
x_n
a_n
T
g(t)
$\frac{1}{T}\mathrm{rect}\left(\frac{t}{T}\right)$
h(t)
f_0
FM
$\sum_{-\infty}^{\infty}\delta(t-iT)$

Figure 2.14. GMSK modulator.

GMSK modulation mainly has two advantages. One advantage is that it has constant envelope [AAS86]. *Therefore, non-linear power amplifiers scarcely impair detection quality. The second advantage is that it can be also represented as a linear modulation technique*[Lau86, JuB92]. *These pulses are detected in the intermediate frequency* $-1/(4T)$*, see Section 4.1. This so-called derotation - the lowpass signal is premultiplied with the complex vector exp(-j2π(t/4T)) - enables linear detection* [Bai90] *and, hence, linear distortions can be mitigated as easily as with linear modulation techniques. An important parameter characterizing the properties of GMSK modulation is the bandwidth-time product BT (b̲andwidth-t̲ime). In GSM, BT is equal to 0.3.*

Mobile radio channel

In radio communication links of terrestrial mobile networks the signal can travel via more than one path from the transmit to the receive antenna. The reason for this multipath environment is that the signal is reflected and scattered by buildings, trees, cars and other obstacles. At the receiver the superposition of many uncorrelated echoes arriving from different directions with different time delays is observed. The suited model is known as the WSSUS (w̲ide s̲ense s̲tationary u̲ncorrelated s̲cattering) model introduced by [Bel63]. The receiver lowpass signal without the additive noise can be described by

$$r(t) = \sum_{m=1}^{M} \sum_{n=1}^{N_m} \alpha_{mn} c(t-\tau_{mn}) e^{2\pi f_{mn} t}, \qquad (2.33)$$

with α_{mn} being the transmission factors, τ_{mn} the path delays and f_{mn} the frequency shifts of the paths. The paths N_m that arrive in a certain delay interval are jointly treated and are called the side paths. The M delay intervals where signal energy is received are called main paths. It is now assumed that all paths in the delay interval arrive at the average time delay τ_m. Thus, (2.33) can be rewritten as

$$r(t) = \sum_{m=1}^{M} c(t-\tau_m) \sum_{n=1}^{N} \alpha_{mn} e^{2\pi f_{mn} t} = \sum_{m=1}^{M} c(t-\tau_m) z_m(t) . \qquad (2.34)$$

The side paths are now united in one path having the complex transmission factor $z_m(t)$ and the mobile radio channel can be modelled by a tapped delay line (see Fig. 2.15).

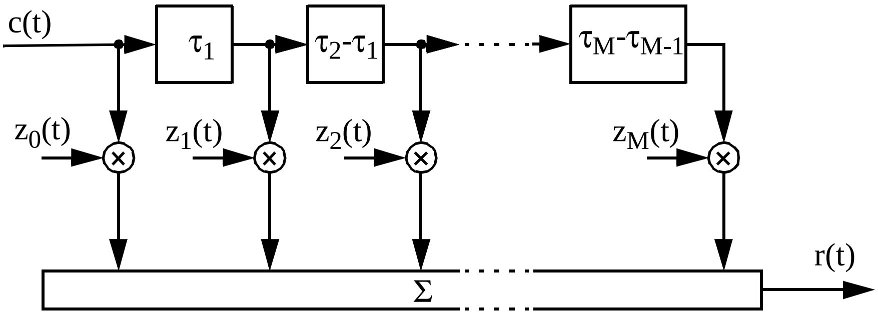


Figure 2.15. Tapped delay line model for the multipath environment

The factors $z_m(t)$ can be large or small depending on whether the side paths add constructively or destructively. At one location the receive signal can be large whereas, in a neighbouring location the signal can be small. A mobile moving through the spatial-varying field observes substantial amplitude fluctuations. This time-variant channel is termed fading. The variables α_{mn}, τ_{mn}, and f_{mn} cannot be described deterministically, and are modelled as random processes. Assuming a large number of side paths N_m, from the central limit theorem it is concluded that the imaginary and the real part of $z_m(t)$ are Gaussian distributed. Hence, the amplitude of $z_m(t)$ is Rayleigh distributed. This type of fading is called Rayleigh fading.

On one hand, the movement in the multipath environment imposes fading. On the other hand, the movement of the mobile user causes frequency shifts (Doppler shifts) of the signal. These shifts depend on the speed of the mobile and on the angle between the incoming path and the direction of the mobile. The Doppler frequency f_D is given by

$$f_D = f_0 \frac{v}{c} \cos\gamma , \qquad (2.35)$$

where v is the velocity of the mobile, c the speed of light, f_0 the carrier frequency and γ the angle of the arriving path. Assuming that the angles of the incoming paths are equally distributed, the classical Doppler spectrum [Jak74] is given by

$$\Phi(\quad) \sim \frac{1}{\sqrt{1-\left(\frac{f}{\ }\right)^{2}}} . \qquad (2.36)$$

Depending on the environment, the received energy is differently distributed. A very common method for characterizing the kind of environment is the power delay profile, the density function of the echoes. A time-discrete expression of the power delay profile is given in [ETS98c] for various environments, e.g. TU (typical urban), RA (rural area), HT (hilly terrain), where the delays τ_m, the average amplitudes of $z_m(t)$ and the Doppler spectra are specified.

Example 2.7 *In COST 207* [COS89] *the power density spectra $q(\tau)$ of various environments have been specified. In Fig. 2.16 the power density function $q(\tau)$ for TU is given, while in Table 2.1 the discrete power delay spectrum as specified in [ETS98c] is listed. From channel measurements* [MoK96] *it is known that the impulse response of the mobile radio channel is not purely continuous. Therefore, the discrete power delay spectrum characterizes the mobile radio channel more exactly.*

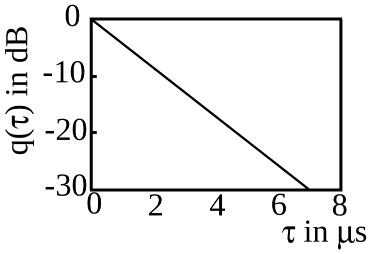


Figure 2.16. Power delay spectrum for TU as specified in [COS89]

Tap number m	relative time delay τ_m(in μs)	average relative power (in dB)
1	0.0	-3.0
2	0.2	0.0
3	0.5	-2.0
4	1.6	-6.0
5	2.3	-8.0
6	5.0	-10.0

Table 2.1. Power delay profile for TU as specified in [ETS98c]

In GSM the Doppler spectrum for each fading tap is specified. For TU all taps have the classical Doppler spectrum, see (2.36).

Receiver noise

At the receive antenna the signal must be amplified. The thermal noise added by the front-end of the receiver can be modelled as a white Gaussian process having the two-sided power density $N_0/2$ [WoJ65].

Receive filtering, downconversion and sampling

Because the signal was modulated to the carrier frequency f_0, at the receiver the signal has to be downmixed to the lowpass domain again. This procedure is normally implemented in various steps via several intermediate frequencies [Viz95]. In this thesis, it is assumed that downmixing is performed in one step, modelled by the multiplication with $\cos(j2\pi f_0 t)$. In order to suppress the frequency components of other users and those which result from downconversion a lowpass receive filter h_{RX} is mandatory. After the receive filter the channel values are sampled at sampling period T_s.

In this thesis symbol-to-symbol MAP equalization [KoB90] is considered. This method of equalization has properties similar to ML equalization (see Section 2.4.2). For optimum ML equalization a whitening matched filter approach must be used [For72, Pro95, Bla90]. However, in mobile radio systems the radio channels are time-varying. Hence, a matched filter has to be time-varying as well. These time variances have to be estimated accurately in order to enable matched filtering. While exact estimation cannot be guaranteed, adaptive matched filtering would introduce side effects. In order to leave these additional effects aside, time-invariant receive filters are used throughout this thesis. The sampling period T_s is equal to the symbol duration T.

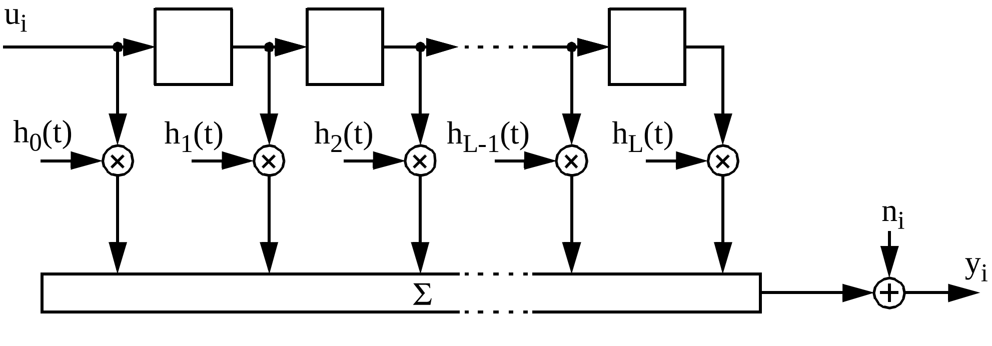


Figure 2.17. Time-discrete ISI channel model.

For linear modulation techniques and linear waveform channels, the entire transmission channel can be modelled with a tapped delay line [Pro95] as depicted in Fig. 2.17. The influences of the modulator and of the radio channel are included within this simple model. As GMSK modulation can only be approximately modelled as a linear modulation scheme, the tapped delay line model is only used for basic simulations. In order to obtain the exact performance of GMSK-based transmission the modulator and the radio channel are modelled separately.

2.4 Equalization

In order to remove the distortions introduced by the transmission channel an equalizer is used.

Channel equalization can be divided into two classes: linear and non-linear algorithms [Pro95]. In linear equalization the detector tries to compensate for the distortions by minimizing the cost function $f_c(\quad)$. For this an adaptive filter can be used. Depending on the type of cost function there exist several detection algorithms, e.g. MMSE (minimum mean square equalizer), zero-forcing block-linear equalizer. Another method that is based on minimizing the cost function is the DFE (decision feedback equalizer). It is a non-linear equalization technique. With respect to error probability all these algorithms are suboptimal. A non-linear equalizer that is optimum with respect to bit error probability is the symbol-by-symbol MAP equalizer [KoB90]. Similar to the decoding algorithms suboptimum derivatives, e.g. the Max-Log-MAP and the SDMA equalizer [MeM92], exist. The SOVA is optimum with respect to sequence error probability. For the same reasons as for the decoding algorithms throughout this work only the Log-MAP and the Max-Log-MAP algorithm are treated.

The trellis-based algorithms model the transmission channel as a time discrete ISI channel model (see Fig. 2.17). Before ISI can be mitigated the channel parameters, i.e. the parameters of the tapped delay line, have to be estimated as they are not known to the receiver. Using the estimated channel parameters the equalizer tries to compensate for the ISI and gives an estimate of the received sequence.

The equalizer can be split into two components. One component is the channel parameter estimator. The second component removes the ISI.

Example 2.8 *The normal burst structure of GSM* [ETS98d] *is illustrated in Fig. 2.18.*

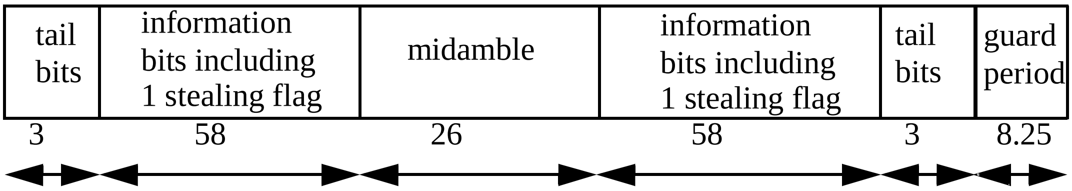

tail bits
information bits including 1 stealing flag
midamble
information bits including 1 stealing flag
tail bits
guard period
3
58
26
58
3
8.25

Figure 2.18. Structure of the normal burst of GSM.

In the middle of the burst the midamble of 26 bits comprising the training sequence is included. At the receiver, the receive values corresponding to the training sequence are processed first. The channel impulse response as well as the signal noise are estimated. Then, the first half burst and the second half burst are equalized separately by using a SISO equalizer. The tail bits are used to terminate the trellis at the edges of the *burst. Because of the delay spread of the radio channels and because of power ramping, a guard period is attached to separate the signals of two consecutive bursts. Note that each information part comprises one stealing flag and 57 user bits.*

2.4.1 Channel parameter estimation

The channel parameters characterising the time-discrete ISI channel model in Fig. 2.17 which must be estimated are:

- the variance, σ^2, of the noise, and,
- the complex channel coefficients, h_i.

In mobile communication systems the channel parameters vary from burst to burst. If methods are used that exploit the diversity of the radio channel, the channel parameters of consecutive bursts are decorrelated, e.g. FH (frequency hopping) [CJL94], antenna hopping [OAH97]. Hence, the channel parameters have to be estimated for each burst.

A robust method enabling reliable estimation is the use of pilot signals embedded in the transmitted data. In TDMA mobile communication systems, e.g. the GSM-system, a midamble comprising a training sequence is embedded in the burst.

At the receiver, the channel values y[n] corresponding to the midamble are observed:

$$y[n] = h[n] * d[n] + n[n], \tag{2.37}$$

with d[n] being the training sequence vector.

The only variables not known to the receiver are the channel impulse response h[n][1] and the noise sequence n[n]. The channel estimator now estimates a channel vector $\hat{h}[n]$ by minimizing a cost function $f_c(\Delta y[n])$ [Kam92] with $\Delta y[n]$ defined by

$$\Delta y[n] = y[n] - \hat{y}[n] = y[n] - (\hat{h}[n] * d[n]) \tag{2.38}$$

being the difference between the received values y[n] and the estimated received values $y[n]$. Assuming white Gaussian noise, a possible cost function is the Euclidean distance. A channel estimator trying to minimize this distance, $\|y[n] - \hat{y}[n]\|^2$, is called the least square estimator.

Example 2.9 *In GSM training sequences are used that have the following autocorrelation function:*

[1] In this chapter the notation h[n] is chosen for the time-discrete channel impulse response. In later chapters, for convenience the notation **h** is used instead.

$$d^*[-n] * d[n] = \delta[n], \tag{2.39}$$

with δ[n] being the unit impulse [MoP92]. *This orthogonality enables easier computation of the channel impulse response, since the least square estimator reduces to a simple correlation estimator* [Kam92]. *The channel impulse response h[n] is estimated by convolving the received values corresponding to the training sequence with the reverse and conjugate training sequence d*[-n]:*

$$\hat{h}[n] = d^*[-n] * y[n] = h[n] * \delta[n] + d^*[-n] * n[n]. \tag{2.40}$$

From (2.40) it becomes obvious that a systematic error $d^*[-n] * n[n]$ *is inherent.*

Having calculated the channel impulse response, $\hat{h}[n]$, the noise variance, σ_N^2, that is equal to the noise power can be determined. This is done by estimating the noise from the receive values corresponding to the training sequence

$$n[n] = y[n] - \hat{y}[n] = y[n] - \hat{h}[n] * d[n]. \tag{2.41}$$

Then, the noise power is determined by

$$\sigma^2 = \frac{1}{N_{tr}} \sum_{n=1}^{N_{tr}} \|n[n]\|^2, \tag{2.42}$$

with N_{tr} being the length of the training sequence minus the memory length L of the channel.

2.4.2 Maximum a posteriori symbol-by-symbol estimation

Having calculated all parameters of the discrete-time ISI channel model, symbol estimation can be performed.

An important property of the transmission channel is that the discrete-time ISI channel can be treated as a convolutional encoder with complex output symbols [BKH97]. The trellis oriented SISO equalizer works similar to a SISO decoder.

The trellis spanned by the channel has the following properties:

- The number of states is M^L, with M being the size of the symbol space.
- Each state has M transitions, e.g. for 8-PSK M is equal to 8.
- The "code" rate of the channel is equal to one. The reference channel symbols are calculated similar to decoding (see Section 2.2): $\hat{c}_{s,0} = \sum_{l=0}^{L} \hat{h}_l x_{i-l}$, $\hat{c}_{s,1} = c_{s,0} - 2\hat{h}_0$, with $\hat{h}_l$ being the l^{th} coefficient of the channel impulse response and $x_{k-L}, \ldots, x_k$ being the transmit bits corresponding to the certain state s.

- In adaptive equalization the trellis modelled by the equalizer becomes time-variant and the reference channel symbols have to be calculated for each trellis stage (see Chapter 5).

Again, the same algorithms as in SISO decoding can be applied. In many mobile communication systems MLSE (maximum likelihood sequence estimation) is used for equalization. Based on the SOVA, the MLSE finds the ML sequence and gives reliability information about the detected symbols. The MLSE is optimum with respect to sequence error probability. Compared to a BCJR-based solution, the difference is small; if no iterative decoding or detection is used, almost the same performance is achieved. Since the topic of this work is turbo-detection, only the MAPSSE (maximum a posteriori symbol-by-symbol estimator) is considered. This algorithm is optimum with respect to symbol error probability and also generates very accurate soft-information. In this work, similar to the channel decoders, soft-output is calculated by the following equation:

$$L(\hat{c}_i) = \ln \frac{\sum_{(s',s),\,c=0} P(S_i = s', S_{i+1} = s, \mathbf{y})}{\sum_{(s',s),\,c=1} P(S_i = s', S_{i+1} = s, \mathbf{y})} . \tag{2.43}$$

The difference from a channel decoder is that instead of the log-likelihood values $L(\mathbf{c})$ the channel values, **y,** are given as input. This affects only the calculation of the logarithms of the transition probabilities, that is, (2.21) and (2.22) are changed:

$$\Gamma_{i,0}(s', s) = \log\{\gamma_{i,0}(s', s)\} = L(\hat{x}_i) - \frac{\left|y_i - \hat{c}_{s',0}\right|^2}{2\hat{\sigma}^2} , \tag{2.44}$$

$$\Gamma_{i,1}(s', s) = \log\{\gamma_{i,1}(s', s)\} = -\frac{\left|y_i - \hat{c}_{s',1}\right|^2}{2\hat{\sigma}^2} . \tag{2.45}$$

The calculation of the transition probabilities is based on the estimated reference values $c_{s',0}$ and $c_{s',1}$ that depend on the estimated channel impulse response $\hat{h}[n]$ and on the estimated noise variance σ. If these estimates are not accurate enough, the calculation of the output L-values suffers and the error performance of the system decreases.

3 General principles of iterative equalization and decoding

Iterative equalization and decoding, also called turbo-detection, was first introduced by Douillard, *et al*. [DJB95] and further examined in [PDG97], [GLL97], and [BKH97]. For transmission channels strongly distorted with ISI, the error performance can be improved significantly. Turbo-detection utilizes the turbo-principle: the component decoders iteratively decode symbols by suing the turbo-feedback/turbo-component to enhance detection reliability. In Section 3.1 the principles of turbo-detection are explained. The component decoders and the turbo-component are described.

In order to apply turbo-detection to GSM, the impacts of this equalization/decoding scheme must be considered. Since most of the traffic in GSM is transmitted via the TCH/FS channel [ETS98b], this chapter begins by observing the effects of this special transmission scheme. Improving the performance of this traffic channel has the highest impact on the system capacity. Two aspects are of interest. First, only a part of the speech frame is convolutional encoded, i.e. the class 1 bits, and a priori information is obtained only for these bits. Second, block diagonal interleaving is used in TCH/FS. Here, a coded frame is interleaved with its previous and its successor frame. Since the code bits of more than one frame are scrambled together, this scheme will be referred to interblock interleaving in this work. In Section 3.2 it is shown that turbo-detection must be modified in order to be applied to this special kind of interleaving. Several schemes are evaluated according to their computational complexity, delay, and error performance. The error performance is given for channels with strong time-invariant ISI. The modulation scheme used is BPSK (binary phase shift-keying). These transmission channels are suited to evaluate the performance of different schemes because iteration gains are significant for conventional turbo-detection.

3.1 The component decoders and the turbo-component

As mentioned earlier, the ISI channel can be regarded as a time-varying convolutional code with complex valued symbols, given by the propagation conditions [BKH97]. Hence, the ISI channel and the channel encoder can be considered as a serially concatenated coding scheme which can be decoded iteratively [SPT95, BeM96d].

In order to equalize and decode the received sequence iteratively, the SISO decoder of Section 2.2 must be modified [DJB95, BDM97a, BKH97]. In addition to the soft-output values $L(\mathbf{u})$ of the information bits **u**, the decoder also must calculate soft-output information $L(\mathbf{c})$ of the code bits **c** (see Fig. 3.1).

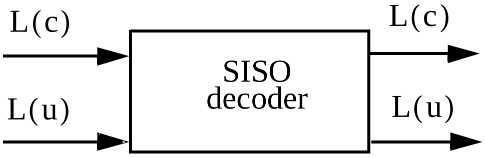


Figure 3.1. Soft-in/soft-out decoder with soft-output information about code bits.

As mentioned in Chapter 2, a MAP decoder is employed. With this algorithm, the information about the code bits can be generated by

$$L(\hat{c}_{i,\mu}) = \ln \frac{\sum\limits_{(s',s),\,c_\mu = 0} P(S_i = s', S_{i+1} = s, L(\mathbf{c}), L(\mathbf{u}))}{\sum\limits_{(s',s),\,c_\mu = 1} P(S_i = s', S_{i+1} = s, L(\mathbf{c}), L(\mathbf{u}))} . \tag{3.1}$$

In (3.1) the probabilities of all transitions that correspond to the same code bit are added and compared for each code bit c_μ of a trellis stage. For example, for a convolutional code of rate R=0.5 the summation and the comparison must be performed twice.

In Fig. 3.2 the scheme of iterative equalization and decoding is depicted. In the 0th iteration the SISO equalizer obtains the channel values **y** and equalizes them. At that time there is no a priori information available from the decoder. If source information, $L^a(\mathbf{x})$, about the code bits **x** is available, it can be supplied to the equalizer. This source information can be obtained for example from the knowledge of the speech encoder statistics. However, in most cases a priori information is only available for the information bits **u**. After the equalizer has calculated the soft-output values, $L^E(\hat{\mathbf{x}})$, the deinterleaved sequence, $L^E_*(\hat{\mathbf{c}})$, is fed to the SISO decoder. As in the 0th iteration there is no a priori information available from the decoder, there is no information extracted before deinterleaving, and the soft-output of the equalizer, $L^E(\hat{\mathbf{x}})$, is equal to $L^E_*(\hat{\mathbf{x}})$. At the decoder a priori information, $L(\mathbf{u})$, about the information bits **u** can be supplied as well. This information again can be obtained from the source encoder statistics using interframe correlation [Hag95] or intraframe correlation [HiR97, HiR98]. The impact of these two algorithms on turbo-detection has been treated in [Wos98] and, hence, is not considered in the following.

The decoder generates two output sequences: the a posteriori values, $L^D(\hat{\mathbf{u}})$, of the information sequence **u**, and the a posteriori values, $L^D(\hat{\mathbf{c}})$, of the code sequence **c**. The 0th iteration is complete. If an estimate $\mathbf{u}$ of the information sequence **u** must be forwarded, only the sign of the soft-values $L^D(\hat{\mathbf{u}})$ has to be evaluated:

$$\hat{u}_i = \begin{cases} 0, \text{ if } L^D(\hat{\ }\ ) \geq 0 \\ 1, \text{ if } L^D(\hat{\ }\ ) < 0 \end{cases}. \tag{3.2}$$

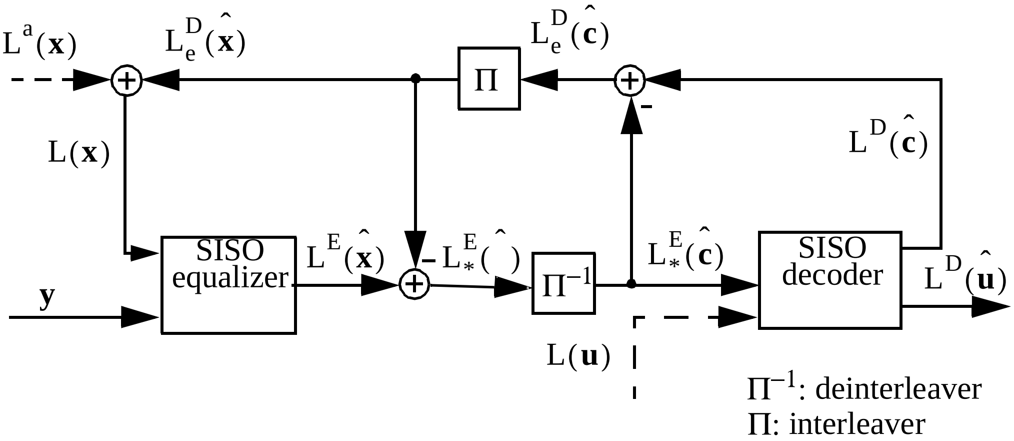


Figure 3.2. Block diagram of iterative equalization and decoding.

In order to improve the bit error rate after the decoder, the same procedure is repeated using the L-values from the decoder for equalization. It is essential to turbo-detection that the decoder does not pass the information to the equalizer that was generated by the equalizer at the previous iteration. Therefore, this information has to be retrieved from the generated a posteriori information at the output of the decoder. This information has two components: the intrinsic part and the extrinsic part. The extrinsic part represents the incremental information generated by the decoder with the information available from all the other bits. This information is fed back to the equalizer. Therefore, the a priori values of the decoder, $L_*^E(\hat{\mathbf{c}})$, representing the intrinsic part, are subtracted from the soft-values, $L^D(\hat{\mathbf{c}})$, to build the extrinsic information, $L_e^D(\hat{\mathbf{c}})$. $L_e^D(\hat{\mathbf{c}})$ denotes the turbo-component of the iterative equalization and decoding scheme. It is interleaved to place the values into the correct order. Another effect of interleaving is that the extrinsic information is decorrelated. After adding the a priori information, $L^a(\mathbf{x})$, the a priori information, $L(\mathbf{x})$, for the equalizer is obtained. The equalizer then obtains the sequence $\mathbf{y}$ and the values $L(\mathbf{x})$. With this additional information the equalizer can generate a more accurate output, $L^E(\hat{\mathbf{x}})$. Before this information is passed to the decoder, the information generated by the decoder at the previous iteration has to be retrieved from $L^E(\hat{\mathbf{x}})$. Therefore, before deinterleaving the equalizer output, the a priori information, $L_e^D(\hat{\mathbf{x}})$, is subtracted. The resulting information sequence, $L_*^E(\hat{\mathbf{x}})$, has then two components: the information from the sequence $\mathbf{y}$ and the extrinsic component which is the incremental information from all other code bits. After deinterleaving, $L_*^E(\hat{\mathbf{c}})$ is fed to the decoder as a priori information. The decoder can now improve the bit error rate by using these improved estimates of the equalizer. In [Kho97] it was shown that if the a priori information $L_e^D(\hat{\mathbf{x}})$ is not removed from the a posteriori values $L^E(\hat{\mathbf{x}})$, the iteration gains decrease significantly, since the L-values passed between the components become too optimistic.

The iterations can now be repeated. From [DJB95] it is known that the iteration gain of the 1st iteration is the largest. The iteration gains decrease from iteration to iteration until the bit error rate performance converges. There are several strategies to limit the number of iterations, such as the stop-criteria. In [BKH97] and [Kho97] several stop-criteria, which adaptively decide when the iterations are stopped, have been evaluated. The authors showed that by adaptively determining the number of iterations for each block, the complexity of the detector can be reduced. If turbo-detection is implemented in receiver boards, it must be ensured that a certain data rate can be processed. Therefore, the receiver components should be designed for the maximum number of iterations. Hence, in practical systems a certain fixed number of iterations can be accommodated, and a trade-off between number of iterations and hardware costs must be found. If there are no restrictions on the energy consumption, e.g., if the energy consumption of the processors of a base station is negligible, there is no need to stop the iterations earlier than at the maximum. For coding schemes where an outer code is used for error detection another strategy can be applied. After each iteration the decoded sequence is checked by the error detector. If the error detector indicates that the decoded sequence is correct, no further iterations are necessary and turbo-detection can be stopped. Particularly for products in which power consumption is a major concern, this strategy seems to be very promising.

3.2 The turbo-detection for inter-block-interleaved GSM transmission

In [PDG97, DJB95, BKH97] it was shown that for channels with severe ISI, e.g. worst case channels [Pro95], large iteration gains are achieved with turbo-detection. In these works the sizes of the coded blocks were chosen to be larger than 4000 bits. With parallel and serial concatenation of convolutional codes [HRP94, BeM969a], the iteration gains strongly depend on the block size [Kho97], that is, the larger the block the larger the gain. In all previous works the entire block is encoded and intrablock interleaved. Intrablock interleaving implies that the data of one code block is not interleaved with the data of another code block. The result of complete coding and intrablock interleaving is that a priori information, $L_e^D(\hat{\mathbf{x}})$, is available for the entire sequence $\mathbf{x}$ after decoding, and that the blocks do not interact with other blocks.

The focus of this work is to apply turbo-detection to GSM-systems. As already mentioned in Example 2.4, in GSM-systems interleaving used for the main services, e.g. TCH/FS, is block diagonal. The code blocks are interleaved with previous and successive code blocks.

Here, the impact of interblock interleaving and delay restrictions of TCH/FS [ETS96] on turbo-detection is examined.

The following two impacts are considered here:

- In GSM TCH/FS only the class 1 bits are convolutional encoded. Therefore, from the SISO decoder a priori information is only available for the coded class 1 bits. For the class 2 bits no a priori information is generated by the channel decoder [ETS98b].
- In [ETS96] maximum delays are recommended. For TCH/FS the time delay imposed by the interleaving and deinterleaving is limited to 37.5 ms; and the time delay required to perform equalization and channel decoding should not exceed 8.8 ms.

If the delay restrictions are obeyed and if it is taken into account that only a part of the speech frame is convolutional encoded, the following result is observed, see Fig. 3.3. Half of the data of frame 0 is transmitted in burst 0 to 3. After 20 ms the speech encoder delivers frame 1. Then, the data to form burst 4 to 7 is available and is mapped on the these four bursts. Every 4.6 ms a burst can be transmitted in a time slot. At least another 18.4 ms are needed to transmit this data. Hence, it takes at least 38.4 ms until the data of frame 0 is completely received. The received bursts are equalized, deinterleaved and frame 0 is decoded. The SISO decoder generates a priori information for the odd positions of burst 4 to 7, and for the even positions of burst 0 to 3. Since only a portion of the data of frame 1 is available, no a priori information is generated for the even positions of blocks 4 to 7. To receive the entire information needed to decode frame 1, another 20 ms would be needed, which is not possible without ignoring the delay restrictions mentioned above.

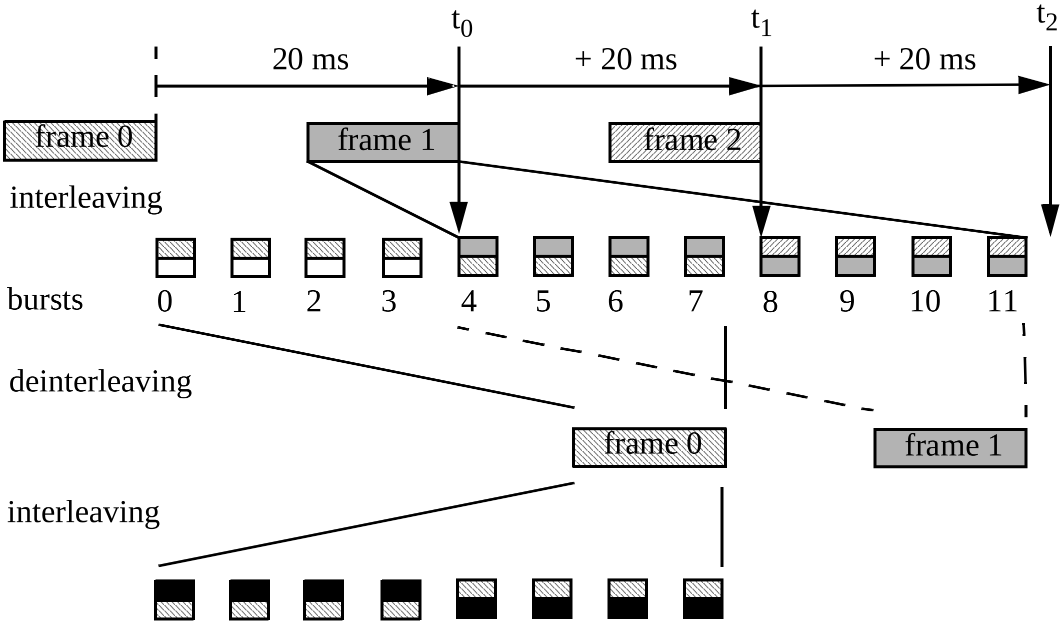


Figure 3.3. The impact of block diagonal interleaving for GSM TCH/FS on turbo-detection.

Furthermore, the decoder generates only a priori information for the 378 code bits of a frame. For the 78 class 2 bits no a priori information is calculated. Incomplete a priori information is obtained because of unequal error protection and the delay restrictions.

Due to this particular coding and interleaving scheme, several possibilities for turbo-detection varying in delay and computational complexity arise. Five schemes are described below.

3.2.1 The original scheme

In order to be able to evaluate the impact of the delay, the delay restrictions are disregarded; this is called the original scheme. Turbo-detection starts after a large number of bursts is received. First, all received bursts are equalized. After deinterleaving, required data to decode all but the last frame is available. Except for the bits of class 2, the complete a priori information for all symbols to be detected is generated. For convenience, the a priori information is regarded as complete.

Having the complete a priori information for iterative equalization and decoding, this scheme gives an lower bound for the error performance of turbo-detection for TCH/FS.

3.2.2 Real-time schemes and its derivatives

The original scheme is not applicable in a real communication system, since the transmission delay would be too large. Particularly for real-time services, schemes have to be considered that exceed the delay only to a certain extent. These real-time schemes vary in delay and computational complexity. In order to gain knowledge about which scheme is to be preferred for implementation, the performance has to be evaluated.

3.2.2.1 Scheme 1 - no additional interleaving delay

Unlike the original scheme, no additional interleaving delay above conventional detection of TCH/FS is introduced in scheme 1.

As shown in Fig. 3.3, no a priori information can be generated for the even positions in the last four bursts that have been received; the information is incomplete. On decoding of frame 0, a priori information for the last eight bursts is generated. There are now two main possibilities for proceeding.

Scheme 1a:

One possibility is to include only the equalization of the last four bursts into the turbo-detection loop. This scheme, denoted as scheme 1a, has the lowest complexity since only the data of the current four bursts and the actual frame is processed. It provides a lower bound for the error performance.

Scheme 1a is depicted in Fig. 3.4. In the 0th iteration, only the channel values $\mathbf{y}$ of the last four bursts are equalized. No a priori information for these bursts is available at that moment. The equalized values, $L_*^E(\hat{\mathbf{x}})$, of these four bursts and the equalized values that have been stored during the iterative equalization of the previous four bursts, are deinterleaved to form the a priori information, $L_*^E(\hat{\mathbf{c}})$, for the decoder. After this sequence is decoded, the extrinsic information, $L_e^D(\hat{\mathbf{c}})$, about the code bits is interleaved again. After interleaving a priori information for the last eight bursts is available. However, in scheme 1a only the values $L_e^D(\hat{\mathbf{x}})$ for the last four bursts are fed back.

After each iteration the a posteriori information, $L^D(\hat{\mathbf{u}})$, about the information bits of frame n are passed to the next receiver unit.

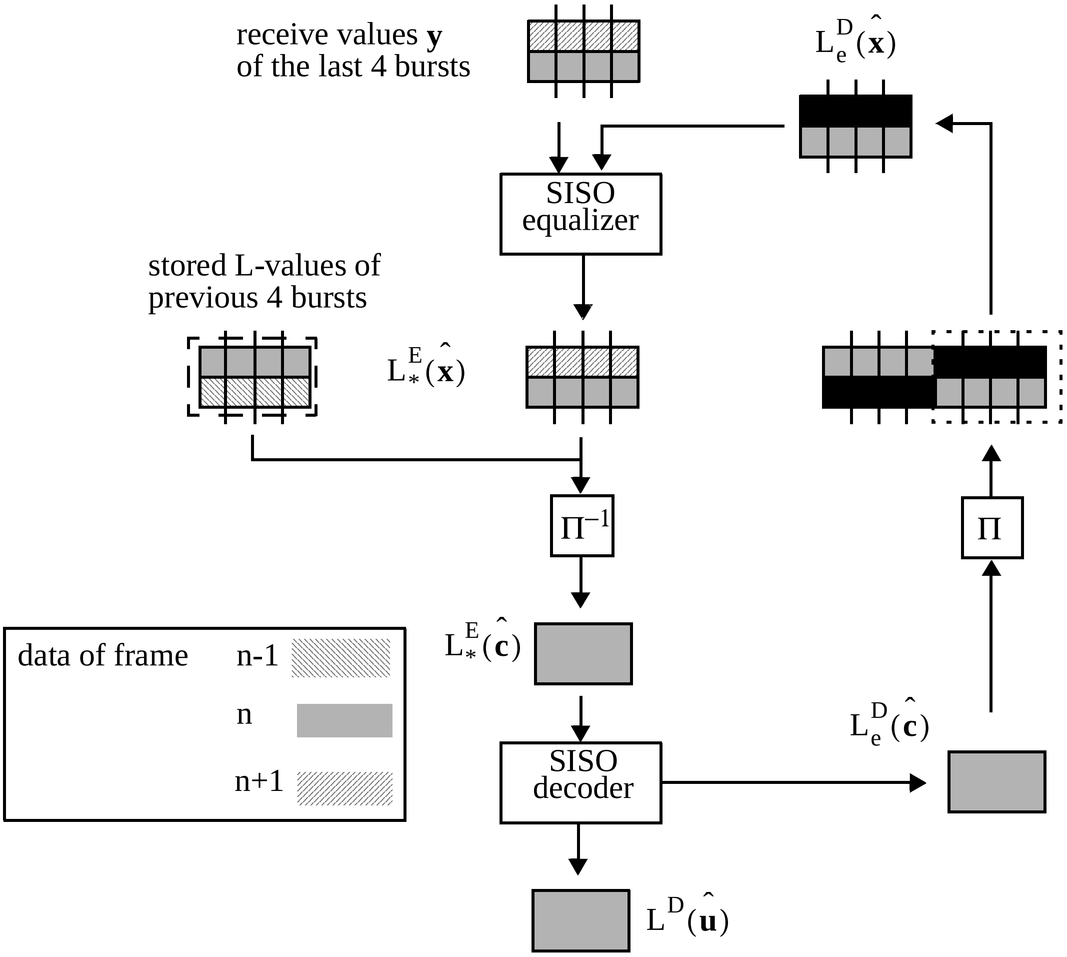


Figure 3.4. Turbo-processing of scheme 1a.

Scheme 1b:

A second possibility is to equalize the last eight bursts during the turbo-detection process. This scheme will be referred as scheme 1b.

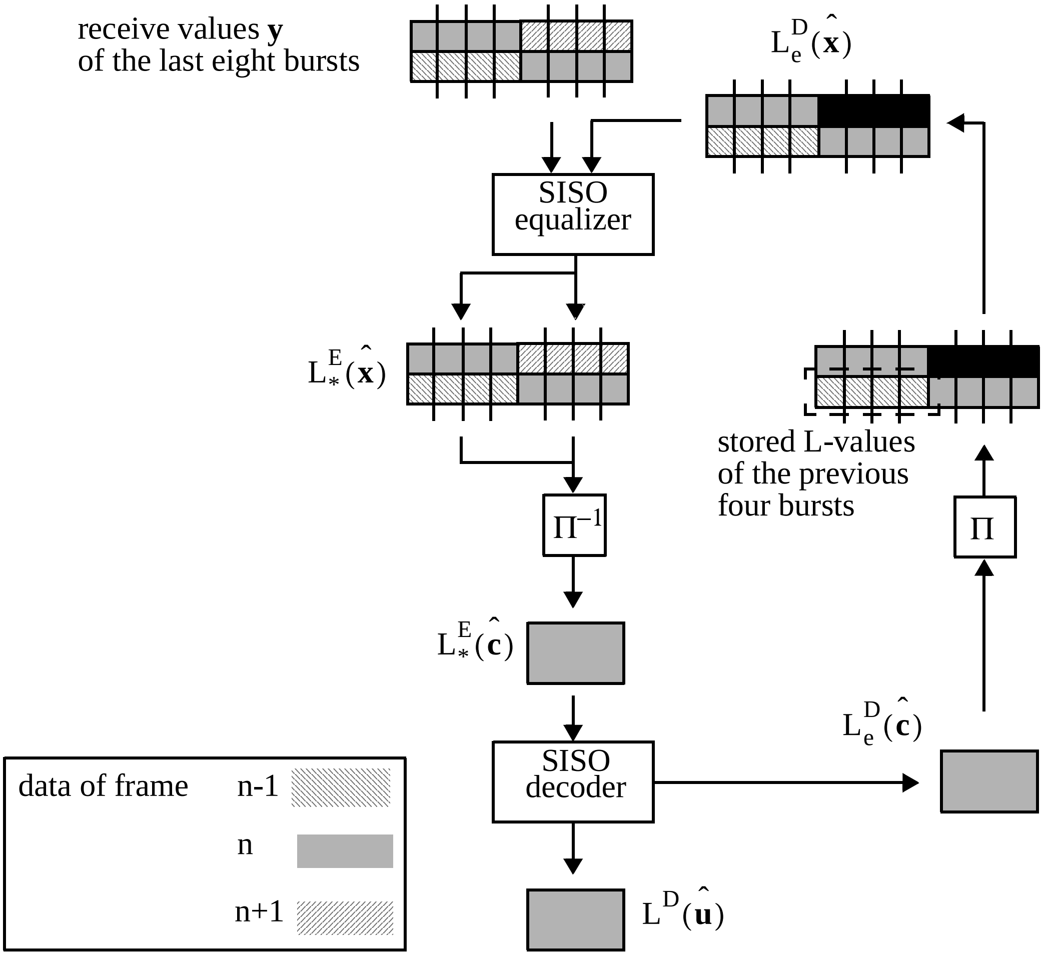


Figure 3.5. Turbo-processing of scheme 1b.

As shown in Fig. 3.5 the values **y** of the last eight bursts are equalized in the turbo-detection process. In the 0th iteration, the equalization of the previous four bursts is not necessary since the values $L_*^E(\hat{\mathbf{x}})$ have been previously calculated during the iterative detection of the previous frame. The data $L_*^E(\hat{\mathbf{x}})$ of the last eight bursts is deinterleaved and the values $L_*^E(\hat{\mathbf{c}})$ of frame n are decoded. The extrinsic information $L_e^D(\hat{\mathbf{c}})$ is interleaved. At the odd positions of the previous bursts the stored L-values from the previous turbo-detection process are inserted. The a priori information, $L_e^D(\hat{\mathbf{x}})$, of the eight bursts is then used for the equalization of the last eight burst. After each iteration the a posteriori information, $L^D(\hat{\mathbf{u}})$, of frame n can be passed to the next receiver unit, e.g., the CRC decoder.

In scheme 1a and scheme 1b, respectively, the L-values $L_*^E(\hat{\mathbf{x}})$ and $L_e^D(\hat{\mathbf{x}})$ from the previous turbo-detection process are utilized. Let us assume that the maximum number of iterations is I. During the turbo-detection of the previous frame n-1 these L-values have been generated I times. The stored values and the values of the current detection of the corresponding iteration can be combined, i.e. on the ith iteration the stored values of the ith iteration of the previous turbo-detection are used. This method of combining is implemented in scheme 1a and scheme 1b.

Another possibility is to use the stored data of iteration I on all iterations. With this modification the L-values of only one iteration are stored. The schemes amended this way are called modified scheme 1a and modified scheme 1b.

3.2.2.2 Scheme 2 - additional interleaving delay of 20 ms

In scheme 1 no additional interleaving delay, compared to conventional GSM processing, was allowed. Therefore, for fewer than half of the values of the last four bursts a priori information can be generated. Allowing an additional delay of 20 ms, a priori information for all the coded bits corresponding to the actual detected frame n can be generated. The processing is depicted in Fig. 3.6.

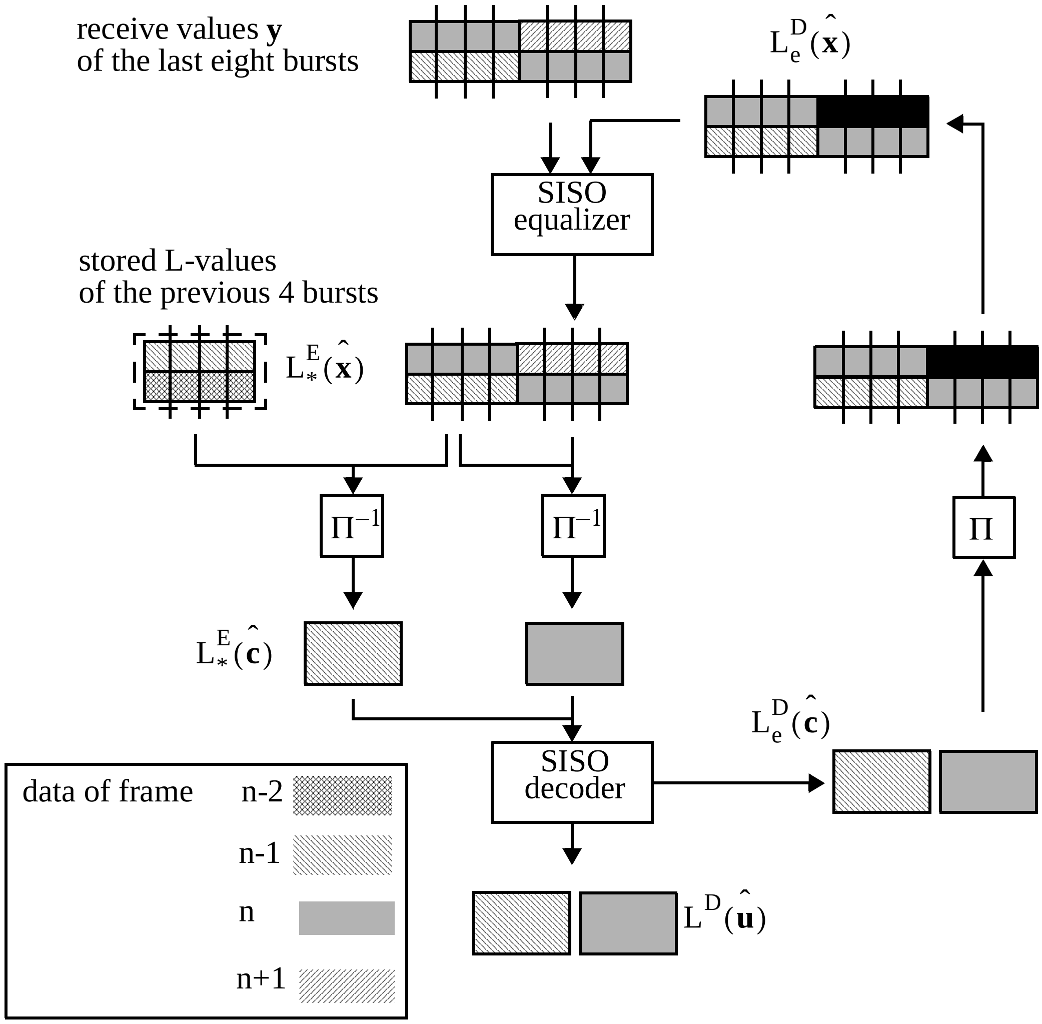


Figure 3.6. Turbo-processing of scheme 2.

In Fig. 3.6 turbo-detection of frame n-1 is illustrated. At the receiver the channel values, **y,** of the last eight bursts are equalized. Then, the information $L_*^E(\hat{\mathbf{x}})$ of the last twelve bursts is deinterleaved. The information of the first four bursts is taken from the turbo-detection of frame n-2. The values $L_*^E(\hat{\mathbf{c}})$ for frame n-1 and n are then available, and both frames are decoded. The extrinsic values, $L_e^D(\hat{\mathbf{c}})$, of both frames are then interleaved and the a priori information, $L_e^D(\hat{\mathbf{x}})$, of eight bursts is obtained. Note that the complete a priori information for four bursts has been generated. After decoding, the a posteriori values, $L^D(\hat{\mathbf{u}})$, of frame n-1 and n can be passed to the next receiver unit. For frame n-1 the complete a priori information is available. The performance is expected to be closer to that of the original scheme.

In scheme 2 the stored values $L_*^E(\hat{\mathbf{x}})$ from the turbo-detection of the frame n-2 are used. Similar to scheme 1, the L-values of the corresponding iterations are combined, i.e., the stored L-values of the ith iteration are combined with the generated values of the ith iteration. The maximum number of iterations is set to I. Again, the stored L-values of the Ith iteration from the detection of frame n-2 for all iterations can be used on all iterations; this scheme is termed modified scheme 2.

3.2.3 Comparison

Different aspects of the equalization/decoding process must be considered when the various turbo-detection schemes are compared. These are

- the computational complexity,
- the memory need,
- the delay, and,
- the error performance.

3.2.3.1 Computational complexity and memory requirement

When evaluating the computational complexity of the various turbo-detection schemes two aspects must be considered.

First, for turbo-detection the generation of accurate soft-output is essential. Hence, equalization and decoding algorithms such as the Log-MAP algorithm or the Max-Log-MAP algorithm, which are more complex than conventional detection, should be used to enable large iteration gains. For non-iterative equalization and decoding it suffices to use the SOVA. The absolute complexity of the algorithms depends on the implementation; however, assuming that all operations, such as max-operations and table look-ups, have the same computational costs, it can be said that the computational complexity of the Log-MAP algorithm is two to three times higher than that of the SOVA [RHV97].

Second, during iterative equalization and decoding the detection processes are repeated several times. The number of equalization or decoding runs depends on the turbo-detection scheme.

To detect one speech frame during conventional reception, four bursts must be equalized and one speech frame must be decoded. These costs are referred as the reference complexity, C_f, per frame. This parameter can be distinguished in the equalization complexity $C_{f,e}$ and the decoding complexity $C_{f,d}$. In the following it is assumed that the equalization and decoding complexities are the same for each iteration[1].

In Table 3.1 the complexity of the various schemes compared to the reference complexity, C_f, is given with I being the number of iterations. Scheme 1a has the lowest computational complexity of all iterative schemes. During each iteration four bursts are equalized, and one frame is decoded. After the 0th iteration the complexity is identical to the reference complexity, C_f. For I iterations the overall complexity is $(\quad) \cdot C_f$. In scheme 1b additional complexity is imposed by the equalization of the previous four bursts. In each iteration, except for the 0th iteration, eight bursts are equalized. During the 0th iteration only four bursts are equalized since the data from the previous four bursts is calculated during the detection of frame n. The equalization complexity is $(\quad) \cdot C_{f,\,e}$. The decoding complexity is equal to that of scheme 1a. In scheme 2 the additional complexity is that, except for the 0th iteration, in comparison to scheme 1b each frame has to be decoded twice. The decoding complexity is given by $(\quad) \cdot C_{f,\,d}$.

Scheme	Equalization complexity $C/C_{f,e}$	Decoding complexity $C/C_{f,d}$
1a	I+1	I+1
1b	2I+1	I+1
2	2I+1	2I+1

Table 3.1. . Relative complexity of the various detection schemes.

The storage requirements for the L-values and the channel values **y** can also be compared. The values are depicted in Fig. 3.4, Fig. 3.5 and Fig. 3.6. Note that the storage needed depends on the word size, w, the memory management and other implementation specific facts such as the board design. To provide approximate figures for the memory required, the storage needed for the channel parameters, stealing flags, midamble, etc., can be neglected.

The length of one coded frame is N bits, e.g., in GSM N is equal to 456. Since the storage of pilot information is not considered, N channel values, **y,** for the equalization of four bursts must be stored. For the equalization of four bursts the soft-output, $L_*^E(\hat{\mathbf{x}})$, and the a priori information, $L_e^D(\hat{\mathbf{x}})$, also have the same length N. The values $L_e^D(\hat{\mathbf{c}})$ and $L_*^E(\hat{\mathbf{c}})$ for one frame are of the same size.

The storage required for the various frames is depicted in Table 3.2. To obtain the storage required for the modified schemes, the terms I·Nw are removed, and 6Nw, 8Nw

[1] During equalization the part of the transition probabilities corresponding to the channel values is only calculated for the 0th iteration. However, since these probabilities have to be stored for all bursts involved in the turbo-detection process of the current frame, the memory requirement increases. This implementation choice depends mainly on the implementation environment. Hence, not all different implementation variations are considered.

and 11Nw for scheme 1a, 1b and 2 are obtained, respectively. The storage need for non-iterative detection is equal to 6Nw. It is identical to that of modified scheme 1a.

Unit	scheme 1a	scheme 1b	scheme 2	non-iterative
y	Nw	2Nw	2Nw	Nw
$L_*^E(\hat{x})$	2Nw+I·Nw	2Nw	3Nw+I·Nw	2Nw
$L_*^E(\hat{c})$	Nw	Nw	2Nw	Nw
$L_e^D(\hat{c})$	Nw	Nw	2Nw	Nw
$L_e^D(\hat{x})$	Nw	2Nw+I·Nw	2Nw	Nw
	(6+I)·Nw	(8+I)·Nw	(11+I)·Nw	6Nw

Table 3.2. . Storage size for the selected detection schemes.

3.2.3.2 Delay

To determine the delay of each scheme, the time until the entire information needed for iterative detection is available is considered. The time for the processing of the algorithms is neglected. Therefore, schemes 1a and 1b have the same delay as non-iterative detection. In scheme 2 an additional delay of 20 ms is introduced since turbo-detection waits until the next frame n+1 is delivered. The delay of the original scheme is orders of magnitude larger than the other schemes, but, as already mentioned, the original scheme is not proposed for implementation.

3.2.3.3 Error performance

Until this point the selected schemes have been compared according to the computational complexity, memory requirements, and delay. The performance of the selected schemes for the TCH/FS are now compared. The error rates at which the various schemes are compared are

- 0.3% for the BER (bit error rate) of class 1, and,
- 3% for the FER (frame erasure rate).

The FER denotes the rate of the not detected erroneous blocks. These error rates are in the same range as the error rates required for GSM TCH/FS [ETS98c]. The ISI channel used in the simulations of this chapters is the time invariant channel with the channel impulse response

$$h[n] = 0.227\delta[n] + 0.46\delta[n-1] + 0.688\delta[n-2] + 0.46\delta[n-3] + 0.227\delta[n-4]$$

also used in [Pro95]. This channel with strong ISI is well suited for the comparison of the various schemes.

In order to examine the effect of inter-block interleaving on turbo-detection, the effects of channel parameter estimation are initially neglected. It is assumed that the channel coefficients and the noise power are known at the receiver. The algorithm used for equalization and decoding is the MAP algorithm.

The original scheme has the best performance of all schemes and provides the bounds to which the real-time schemes are compared. In Fig. 3.7, Fig. 3.8 and Fig. 3.9 the BER of the class 1 bits after the decoder, the FER, and the raw BER up to 9 iterations are depicted, respectively, with E_b being the energy per modulated bit. The iteration gains are emphasized for the 1st, the 3rd and the 9th iteration. Note that the raw BER denotes the BER after the equalizer.

In Fig. 3.7 it can be seen that on the 1st iteration the iteration gain for the BER of class 1 bits is approximately 2 dB. From the 1st to the 3rd iteration an additional gain of 1.2 dB is obtained. The next five iterations improve the BER of the class 1 bits by another 0.6 dB. The BER has already converged after the 9th iteration. No significant improvement can be obtained by further iterations. The overall gain achieved by turbo-detection is about 3.8 dB at a BER of 0.3% for the decoded class 1 bits. Note that all simulations given throughout this thesis have been performed in 0.5 or 1 dB steps. The measured points are connected without applying interpolation techniques. To highlight this, the crosses in Fig. 3.7 and in Fig. 3.8 indicate the measured points. For symplicity they are not given in other figures.

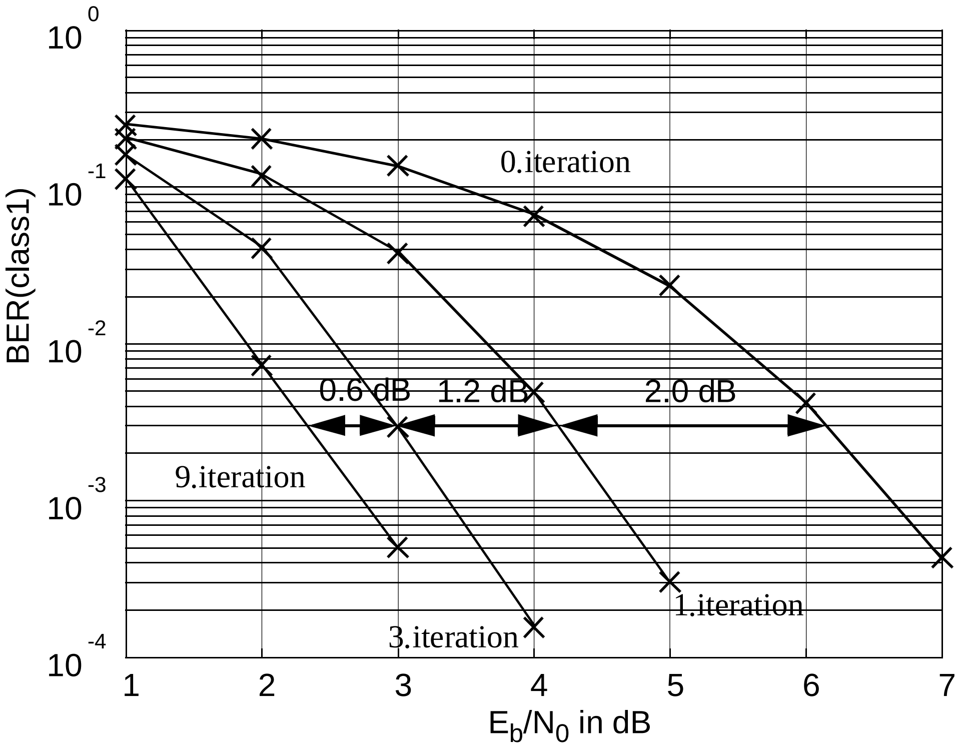


Figure 3.7. BER (class 1) of original scheme (no delay restirctions).

The same behaviour can be observed for the FER. Note that the FER is measured by the BFI (bad frame indicator) of the block decoder [ETS98b]. The iteration gains are 2.1 dB, 3.3 dB and 4.1 dB after the 1st, the 3rd and the 9th iterations, respectively (see Fig. 3.8).

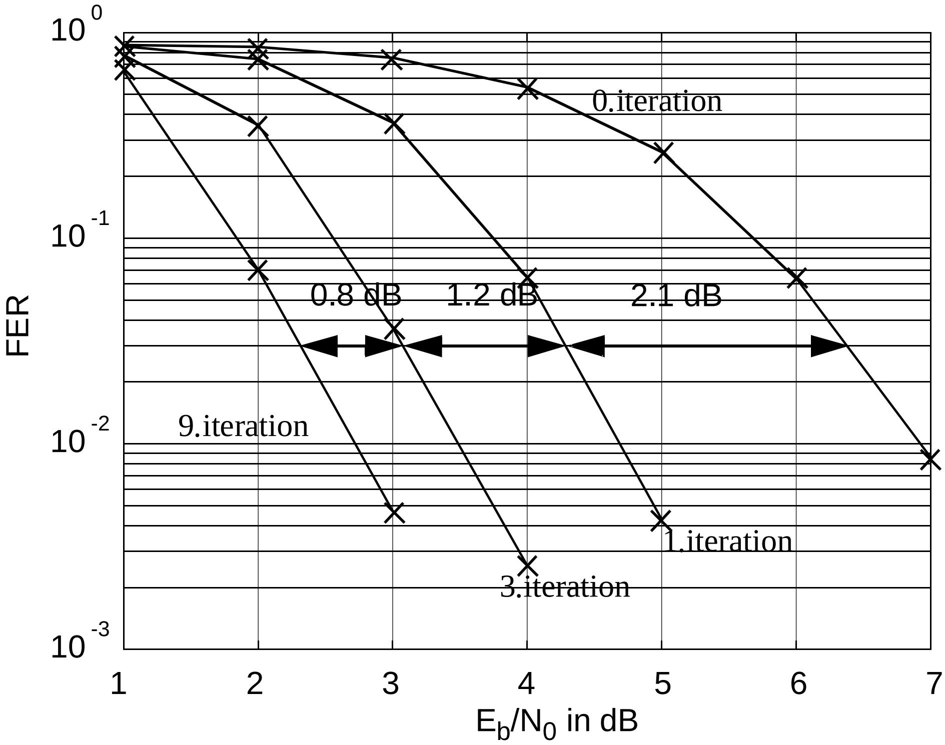


Figure 3.8. FER of original scheme (no delay restrictions).

For the raw BER, i.e. the BER after the equalizer, the behaviour is different. Here, the iteration gains are higher than those for the error rates after the decoder, e.g. the iteration gain of the 1st iteration is about twice as high as for the other error rates. The iteration gains are 3.5 dB, 4.9 dB, and 5.5 dB after the 1st, the 3rd, and the 9th iteration, respectively, at a raw BER of 8% (see Fig. 3.9). The reasons for this behaviour are that the code constraints and the interleaving gain are used to improve the raw BER. On the 0th iteration each burst is equalized without any information from other bursts.

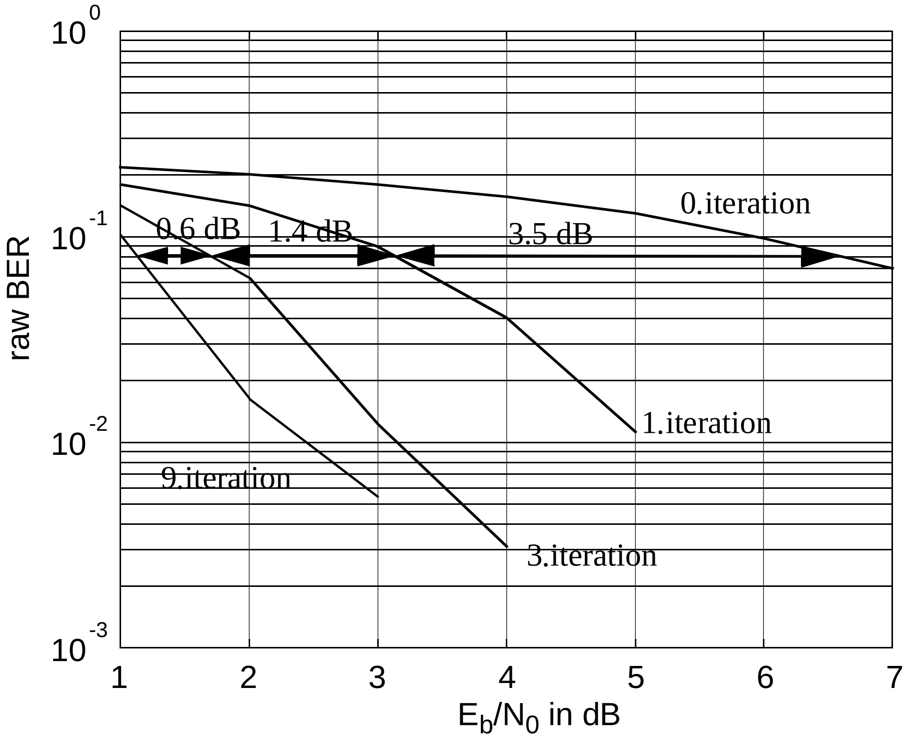


Figure 3.9. Raw BER of original scheme (no delay restrictions).

In Fig. 3.10 the BER(class 1) performance of scheme 1a is given for various iterations. The BER of the 9th iteration is compared to that of the original scheme. The error bars given depict the 95% confidence interval: $P(|h - p| < \varepsilon) = 0.95$ with h being the relative frequency, p being the error probability, and ε being the allowed derivation. The method for the calculation of these intervals is given in [Rie98b]. The variance and the mean of the measured relative frequency are recursively calculated until a certain confidance is reached. Especially for correlated fading channels it is necessary to extend the simulation period to ensure enough statistical independant error events. If ideal frequency hopping is used, the channel impulse responses are independent from burst to burst and reliable results are obtained after a smaller period. Note that throughout this work, if given, the error bars always represent the 95% confidence intervals.

The iteration gains of scheme 1a are, as was expected, lower than those of the original scheme; the achievable turbo-detection gain is 2 dB. After the 3rd iteration no additional iteration gains can be achieved; and after the 9th iteration the BER performance is about 1.8 dB worse than that of the original scheme, i.e. the iteration gain is approximately half that of the original scheme. The same tendency was observed for the FER and for the raw BER.

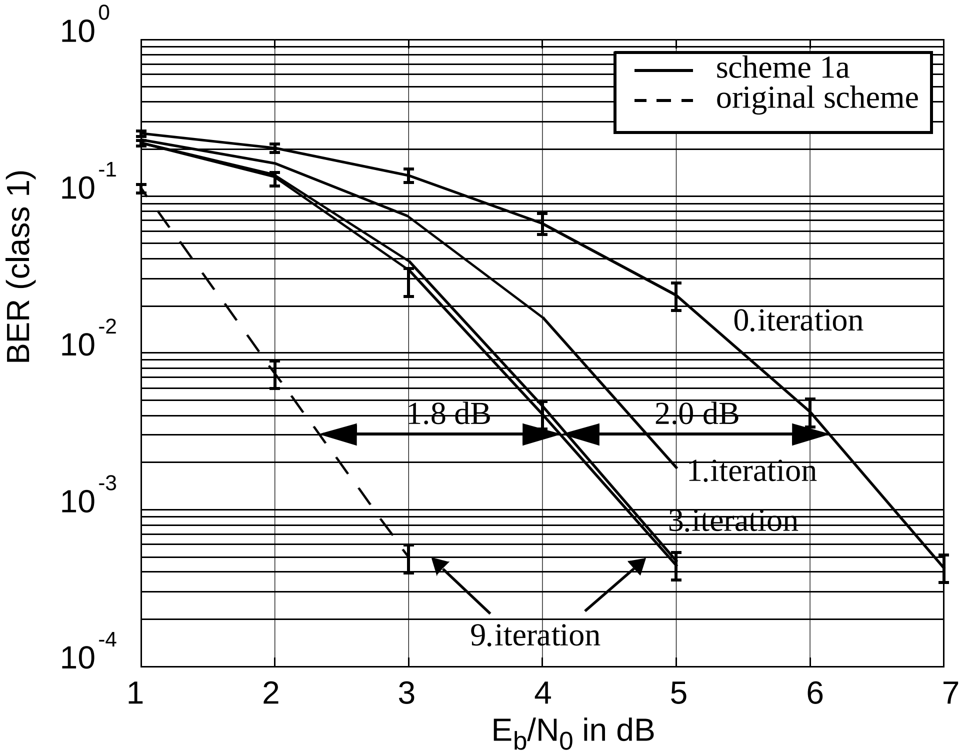


Figure 3.10. BER(class 1) of original scheme and scheme 1a

As mentioned in Section 3.2.2, there are possible modified versions of the detection schemes 1a, 1b, and 2. In the modified schemes the information from the final iteration of the last turbo-detection process is taken. Therefore, these schemes have lower memory costs than the non-modified. The performance of the modified scheme 1a is depicted in Fig. 3.11. Three properties of the modified schemes[1] are observed:

- First, the performance after the 0th iteration is better for the modified scheme, see Fig. 3.11. The reason for this is that during the 0th iteration the information of the highest iteration of the last turbo-detection process is utilized.
- Second, from Fig. 3.10 and Fig. 3.11 it can be seen that the modified version converges faster than the non-modified version. After the 1st iteration the BER has already converged to its final value. No additional gains are achieved for further iterations which implies that the decoder output after only the 1st iteration can be passed to the data sink without any degradation of the error performance. However, other iterations still have to be processed to prepare the L-values for the detection of the next frame.

[1] Note that the same properties apply to the other modified schemes as well. This has been confirmed from simulations. The simulation results are not presented here.

- Third, the error performance of the modified and non-modified schemes converge to the same value. This has two positive effects. On the one hand, less memory is needed. Therefore, the modified scheme is easier to implement. On the other hand, the maximum number I of iterations is decisive; one simulation run with the non-modified scheme suffices to obtain the error rates for selected values of I, i.e. the error rates of the non-modified scheme for the Ith iteration are the error rates the modified version will converge to if the maximum number of the iterations is I.

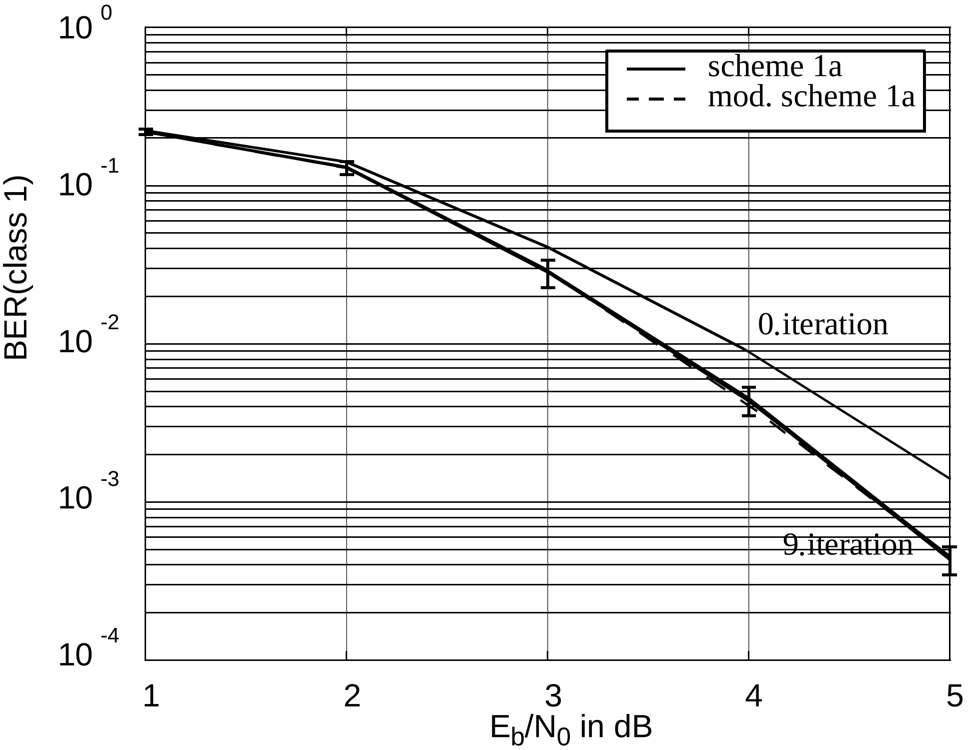


Figure 3.11. BER (class 1) of scheme 1a and its modified variation.

In Fig. 3.12 the BER(class1) after the 9th iteration for the various turbo-detection schemes are compared. The performance of scheme 2 is close to that of the original scheme; the difference is only 0.5 dB. The schemes 1a and 1b perform worse because within these detection schemes a priori values for only half of the code bits are obtained. The degradations for scheme 1b and 1a are 1.5 and 1.8 dB, respectively.

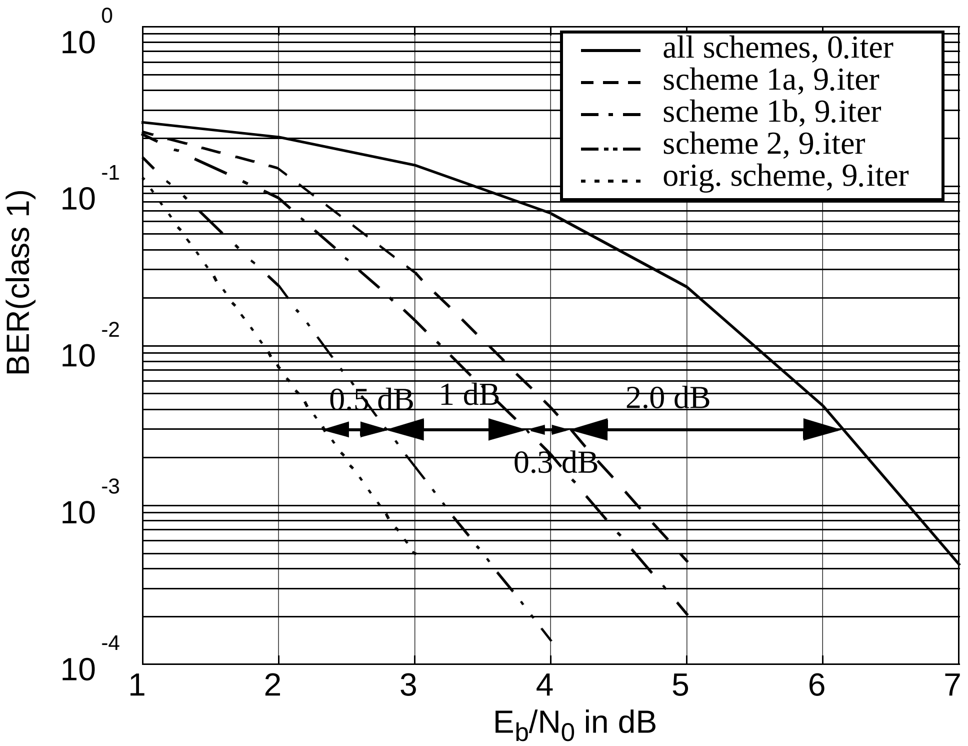


Figure 3.12. BER(class 1) for all turbo-detection schemes after the 9th iteration.

Scheme 1a has the worst performance; however, still an iteration gain of 2.0 dB is achieved. At the cost of additional computational complexity the performance can be improved. After nine iterations scheme 1b outperforms scheme 1a by about 0.3 dB. The equalization complexity has doubled, see Table 3.1. By allowing an additional delay of 20 ms and by approximately doubling the decoder complexity the turbo-detection gain can be improved by 1 dB. The ‘ideal’ performance of the original scheme can be approached as close as 0.5 dB.

In Table 3.3 the BER(class 1) and the FER performance of the real-time schemes are compared to the performance of the original scheme. The turbo-detection gains are compared after one, three, and nine iterations to the iteration gains of the original scheme. As mentioned above, these gains are smaller; hence, the values are negative. In addition the absolute gains of the original scheme are given. By adding the differences to the absolute gains of the original scheme, the absolute gains for the various schemes can be obtained. It can be observed that scheme 2 approaches the performance of the original scheme. After one iteration no difference is observed. For higher iterations the discrepancy increases to 0.5 db for the BER (class 1) and to 0.3 dB for the

FER. By introducing an additional delay of 20 ms compared to state-of-art detection in GSM nearly the same gains as for the original schemes can be achieved.

Difference to original scheme in dB		**Scheme**			**Absolute gain of orig. scheme in dB**
		1a	**1b**	**2**	
1st iteration	BER (class 1)	-0.6	-0.5	0	2.0
	FER	-0.6	-0.5	0	2.1
3rd iteration	BER (class 1)	-1.2	-0.9	-0.2	3.2
	FER	-1.2	-0.8	-0.3	3.3
9th iteration	BER (class 1)	-1.8	-1.5	-0.5	3.8
	FER	-1.7	-1.3	-0.3	4.1

Table 3.3. . BER and FER performance of the various schemes compared to the original scheme

For the schemes where no additional delay is accepted the degradation is larger. The turbo-detection performance converges after three iteration, see Fig. 3.10. Hence, degradation increases with the number of iterations. After one iteration the discrepancy amounts to about 0.5 dB where after nine iterations the divergence in performance comes to the range of 1.5 dB.

Up until this point the behaviour of turbo-detection without the effects of channel parameter estimation has been examined. The channel impulse response h[n] as well as the noise power σ^2 were known to the receiver. In a real system the channel parameters are not known to the receiver, so that the influence of mismatched detection has to be treated. The impact of channel parameter estimation on the various turbo-detection schemes is investigated. The channel parameters are estimated for each frame according to Section 2.4.1.

In Fig. 3.13 the FER for the original scheme is shown with and without ideal channel knowledge after the 0th and 9th iterations. For the 0th iteration the imperfect channel knowledge causes a degradation of 2.2 dB. After the 9th iteration the difference between the performance with and without ideal channel knowledge has amounted to 3.5 dB because the turbo-detection gain is higher if the channel parameters are perfectly known to the receiver. With channel estimation the turbo-detection gain for nine iterations is 2.8 dB. This is 1.2 dB less than for turbo-detection with ideal channel knowledge.

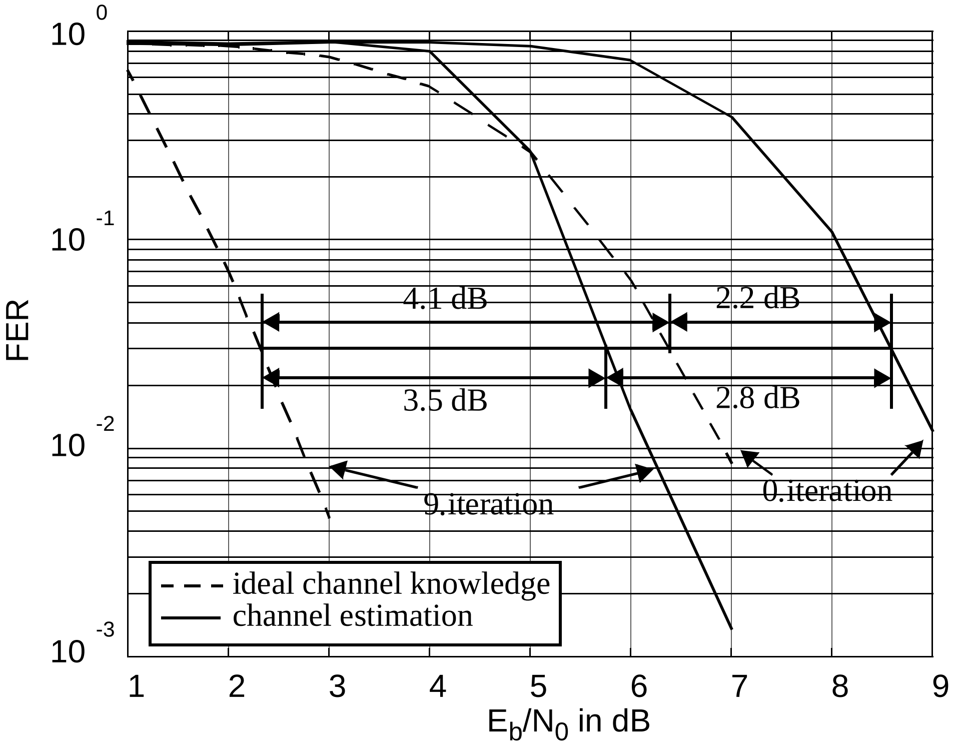


Figure 3.13. FER of original scheme for ideal channel knowledge and channel estimation.

In Table 3.4 the BER(class 1) and the FER performance of the real-time schemes are compared to the performance of the original scheme for channel estimation. The turbo-detection gains are compared after one, three, and nine iterations. Also the absolute gains of the original scheme are given. By adding the differences to the absolute gains of the original scheme the absolute gains for the various schemes can be obtained.

Difference to original scheme in dB		**Scheme**			**Absolute gain of orig. scheme in dB**
		1a	**1b**	**2**	
1st iteration	BER (class 1)	-0.4	-0.3	0	1.4
	FER	-0.3	-0.3	0	1.3
3rd iteration	BER (class 1)	-0.7	-0.5	-0.1	2.2
	FER	-0.6	-0.5	-0.1	2.2
9th iteration	BER (class 1)	-1.1	-0.8	-0.3	2.7
	FER	-1.0	-0.8	-0.3	2.8

Table 3.4. . BER and FER performance of the various schemes compared to the original scheme when channel estimation is used.

The same relations between the selected schemes can be observed as for turbo-detection with ideal channel knowledge. Scheme 2 again is closest to the original scheme and scheme 1a has the worst performance. However, as the absolute iteration gains of the original scheme are not as high the difference between the various schemes also decreases.

In Fig. 3.13 it was shown that due to imperfect channel knowledge, detection degrades significantly. As mentioned previously, the calculation of the equalizer soft-output becomes inaccurate because the reference metrics deviate from the correct reference metrics and because the estimated noise power also influences the APP calculation. As shown in [BaF98b], the influence of the noise power estimate is reduced if using the suboptimal Max-Log-MAP algorithm. Hence, if the channel estimates are not ideally known the suboptimality of the Max-Log-MAP algorithm only has a small impact on the error performance as shown in Fig. 3.14. After the 0th and 9th iteration the performance is about the same for both algorithms.

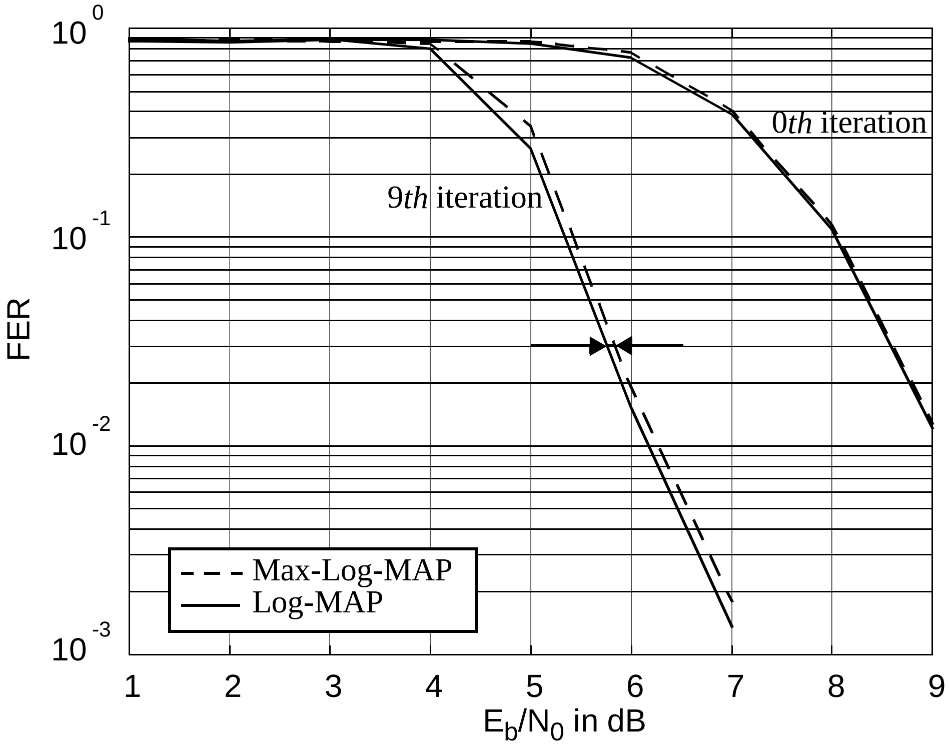


Figure 3.14. FER of original scheme with channel estimation using Log-MAP or Max-Log-MAP algorithms for equalization and decoding.

In Fig. 3.15 the error performance for ideal channel knowledge is depicted. Here, the impact of the suboptimum Max-Log-MAP algorithm is larger than for the case of mismatched detection; however, the difference between the suboptimum and optimum detector is only small.

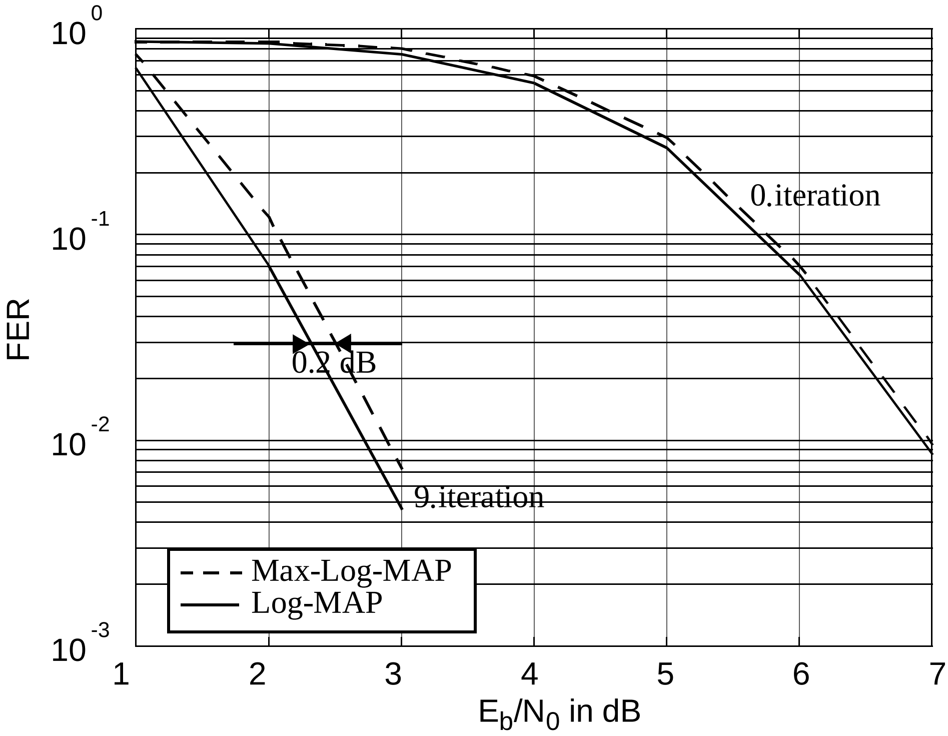


Figure 3.15. FER of original scheme with ideal channel knowledge using Log-MAP or Max-Log-MAP algorithms for equalization and decoding.

If using other suboptimum algorithms, e.g. the SOVA, the degradation is higher [BaF98b].

4 Turbo-detection for various modulation techniques

In Chapter 3 the impact of the coding and interleaving scheme of the GSM TCH/FS was investigated. For interblock interleaving real-time schemes were presented and it was shown that on transmission channels with strong ISI large iteration gains are obtained with these turbo-detection schemes.

In mobile radio systems the ISI is determined by the modulation scheme and the delay spread of the radio channel. Except for the equalizer test channel, the signals transmitted over the specified mobile radio channels experience only small delay spreads, i.e. most paths arrive within one symbol duration T (T=3.69 µs). Hence, the delay spread of the mobile radio channel imposes only little ISI. Strong ISI on the transmission channel mainly origins from the modulation scheme.

This chapter shows the influence of the modulation techniques on the ISI and its impact on turbo-detection. In Section 4.1 the GMSK modulation is treated in combination with Rayleigh fading channels. GSM TCH/FS as well as GPRS transmission is examined. Section 4.2.1 explains how turbo-detection is modified for higher-order modulation techniques, i.e. M-ary modulation. In Section 4.2 turbo-detection is applied to EDGE services where 8-PSK modulation is used.

4.1 Turbo-detection for GMSK modulation

4.1.1 The full-rate speech traffic channel

For error performance evaluation of the various turbo-detection schemes the entire transmission channel has been modeled as time-invariant and distorted with severe ISI. The power of the main signal was lower than that of the delayed interfering signals. In order to gain more expertise about the benefits of turbo-detection for GSM-systems, the transmission channel used for the simulations has to model the real environment. Hence, a GMSK modulator with BT=0.3 as specified for the full-rate speech traffic channel and a fading channel are used in the following.

in real environments, simulations which exactly model the modulator and the fading channel preferred.

In the following, the investigations for interblock interleaving as described in Chapter 3 are completed, and the influence of the GMSK modulator is examined. Again, the TCH/FS coding scheme, which is the most frequently used, and is hence the most capacity consuming channel, is treated here.

In [ETS98c] several mobile radio channels are specified. Since all their impulse responses exhibit relatively small delay spreads, it suffices to begin inquiries into the effects of turbo-detection for one Rayleigh fading channel such as the TU50 channel with ideal FH[1].

In Fig. 4.1 the FER for TCH/FS transmitted via a TU50 channel with ideal FH are depicted. The FER are given for scheme 2, the original scheme, and for scheme 1b. Results for scheme 1a are not given because the iteration gains of scheme 1a are smaller than those of scheme 1b, as shown in Section 3.2.3.

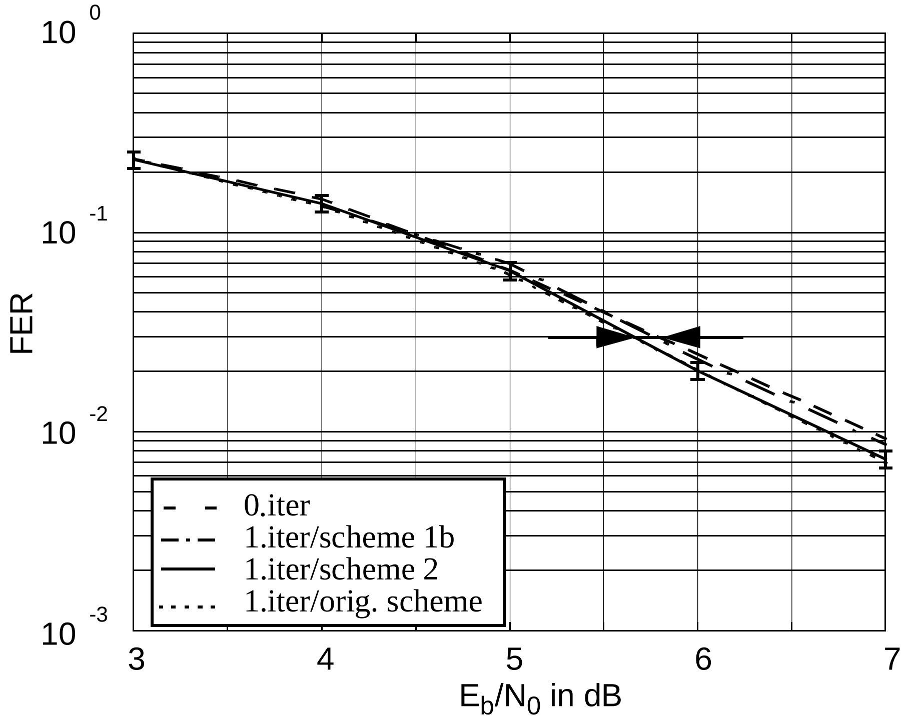


Figure 4.1. FER for TCH/FS on TU50 channel with ideal FH for original scheme, scheme 1b and scheme 2.

From Fig. 4.1 the following conclusions are observed:

- By applying the original scheme, a gain of approximately 0.15 is realized. However, this gain is within the inaccuracy of the simulation. Note that the results of other iterations are left out for clarity as they are nearly identical to the results of the 1st iteration. The gain is already obtained during the 1st iteration; the error performance has already converged and no further gains can be achieved with more iterations.
- The performance of the original scheme can be approached using scheme 2. The additional delay of 20 ms, and the connected turbo-detection with "complete" a priori information guarantees this similar iteration gain.

[1] In ideal FH it is assumed that the distortions of the mobile radio channel are uncorrelated from burst to burst, i.e., the channel coefficients are drawn anew for each burst according to the statistical properties of the channel.

- Without the additional delay even these small iteration gains of scheme 2 and the original scheme cannot be preserved. By using scheme 1b no iteration gain remains.

In the results of Fig. 4.1 the FER is calculated independently from the BFI (bad frame indicator) of former iterations. Even if the CRC decoder has detected a block as correct during former iterations, the data of the current iteration is evaluated. Due to the fact that the data of at least two consecutive frames is processed during turbo-detection for TCH/FS, it is probable that during the iterations a block already detected as correct is falsified. In order to prevent this worsening effect, it is possible to forward the data corresponding to the first correctly detected frame. As already mentioned in Section 3.1, the information block can be passed to the data source if the block decoder indicates that the block is correct.

Applying this principle, larger iteration gains are achieved, see Fig. 4.2. At a FER of 3% a gain of 0.3 dB and 0.5 dB is obtained with schemes 1b and 2, respectively. Iteration gains are only observed during the 1st iteration. The iteration gains of the 2nd iteration are negligible.

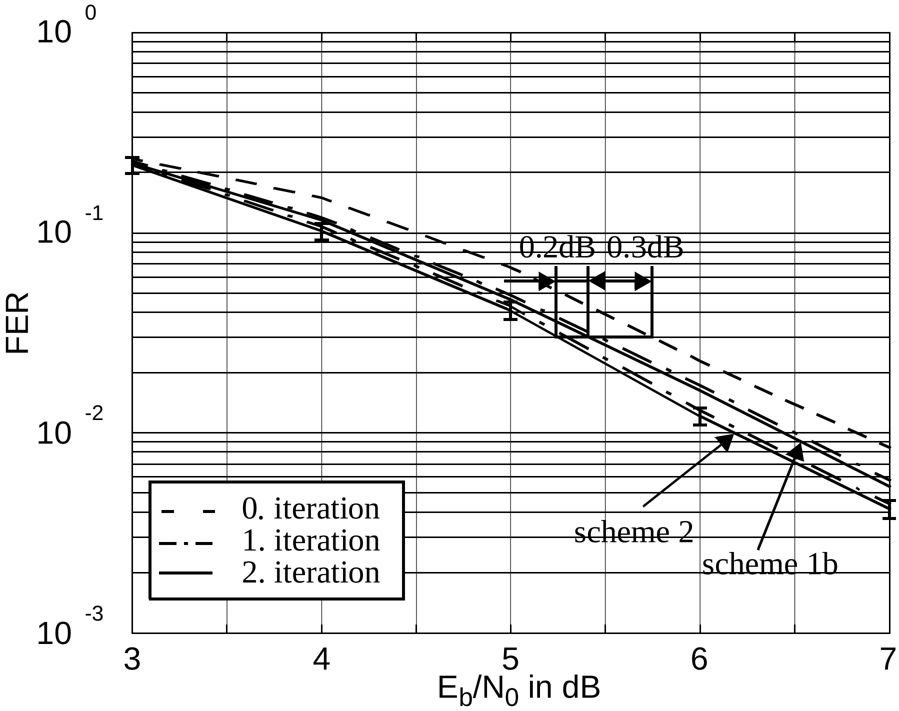


Figure 4.2. FER for TCH/FS on TU50 channel with ideal FH for original scheme, scheme 1b, and scheme 2 forwarding the first correctly detected frame.

The results for the BER (class 1) after the decoder are depicted in Fig. 4.3. Similar gains can be observed for the BER (class 1) as in Fig. 4.2. By applying scheme 2, an iteration gain of 0.5 dB is achieved on the 1st iteration at a BER of 0.3%. Again, the gain amounts to only the half if scheme 1b is utilized.

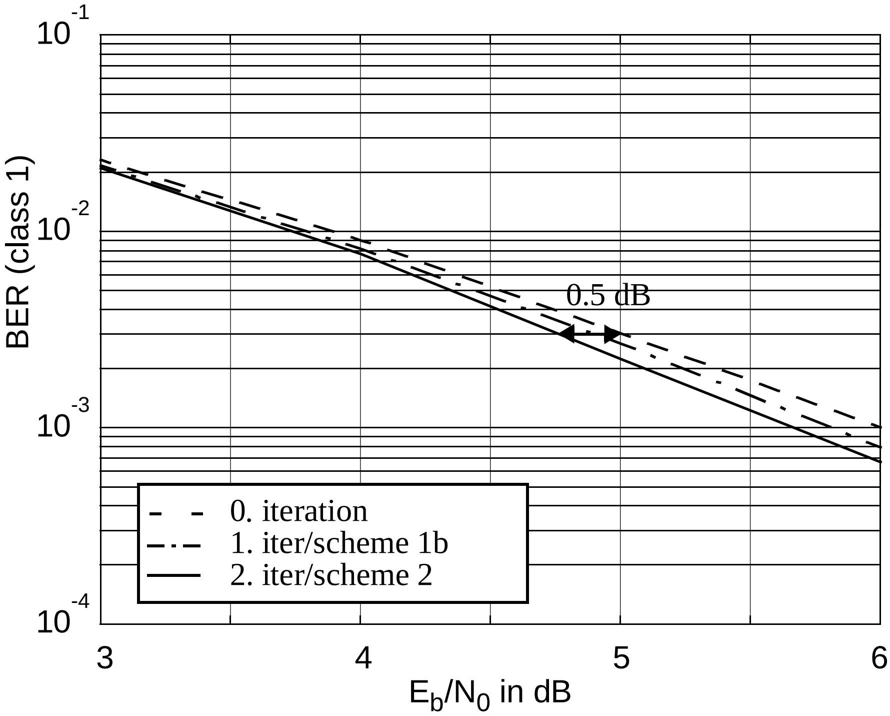


Figure 4.3. BER (class 1) after the decoder for TCH/FS on TU50 channel with ideal FH for original scheme, scheme 1b, and scheme 2 forwarding the first correctly detected frame.

In Fig. 4.4 the raw BER for the schemes 1b and 2 are depicted after the 0th and the 1st iteration, respectively. The results for the 2nd iteration are not given as they are nearly identical to the results of the 1st iteration. Compared to the FER performance and the BER performance after the decoder, the improvement of the BER is significant due to the fact that the “raw BER versus E_b/N_0”-graph is flatter. Hence, any slight improvement in the BER after the decoder results in a larger gain after the equalizer. A gain of about 2 dB is already achieved using scheme 1b. With scheme 2 this improvement is even higher.

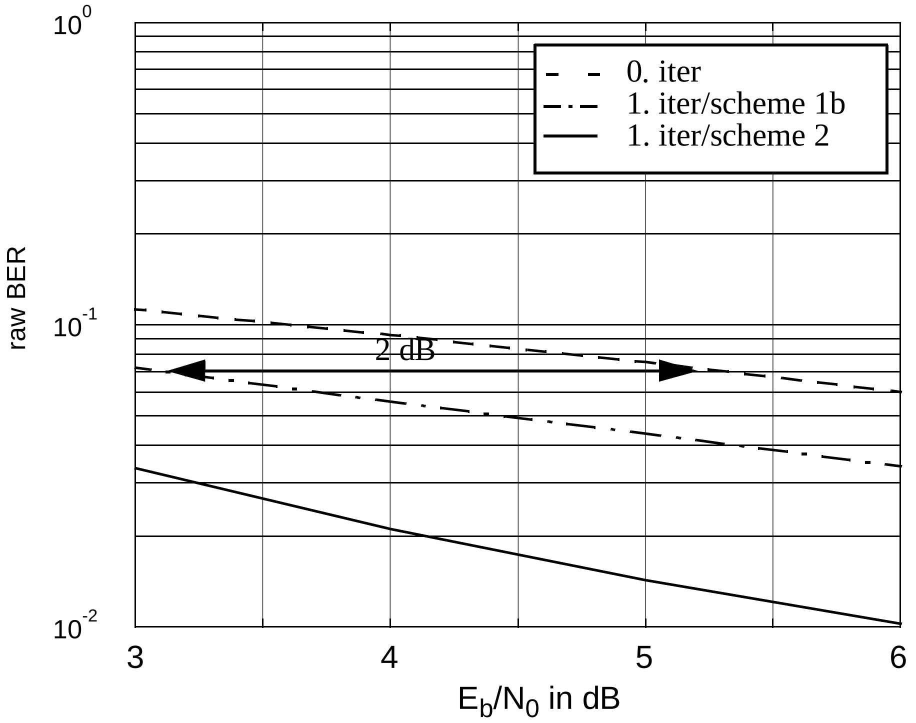


Figure 4.4. Raw BER for TCH/FS on TU50 channel with ideal FH for scheme 1b and scheme 2.

In Section 3.2 large iteration gains have been given. For GMSK modulation and the time-variant fading channel, that is, the TU50 channel, the gains after the decoder dwindle or disappear. To explain this performance, the ISI inherent from the modulator has to be examined. As already mentioned in Section 2.3, GMSK modulation can be also interpreted as the superposition of amplitude modulated pulses [Lau86, JuB92, AAS86]. In Fig. 4.5 the inphase and quadrature components of a GMSK modulated signal (BT=0.3) are illustrated. Each bit is convolved with the impulse response g(t). As shown in Fig. 4.5, the bits are alternatively sent in the inphase and quadrature component. The impulse responses of two consecutive bits overlap to a large extent. However, the interference imposed is orthogonal. The largest non-orthogonal contribution to the ISI stems from the second to last bit. For BT equal to 0.3 the signal impulse response has nearly faded after two symbol periods.

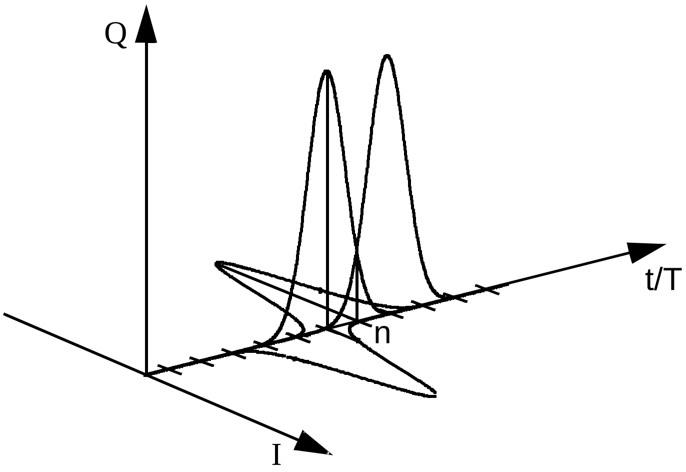


Figure 4.5. Inphase and quadrature component of a lowpass GMSK signal.

For coherent detection of GMSK signals, lowpass processing is performed in the intermediate frequency -1/4T [Bai90]. In the intermediate frequency domain, the impulse response g(t) is multiplied with the complex vector $\exp(-j2\pi(t/4T))$. This denotes the so-called derotation. The resulting impulse response $h(t) = g(t)\exp(-j2\pi(t/4T))$ is depicted in Fig. 4.6.

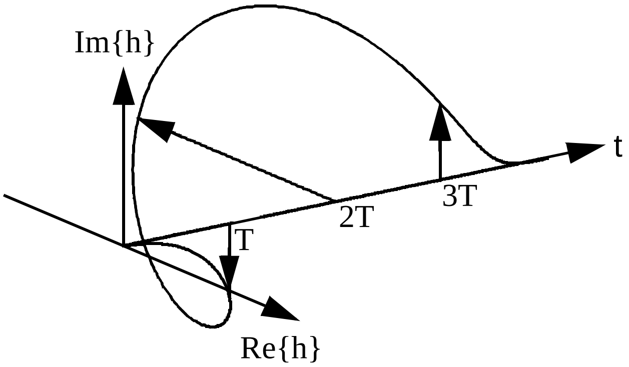


Figure 4.6. Impulse response in the intermediate frequency domain.

This impulse response is illustrated in Fig. 4.6. Two consecutive samples at a distance of one symbol period T are orthogonal.

Now, the poor performance of turbo-detection for GMSK modulated signals transmitted via the specified mobile radio channels can be explained. Only the weak ISI contribution of the bit n-2 and bit n+2 to bit n is used to iteratively improve the decoder error rates. Even the ISI of the mobile radio channel, e.g. the TU-channel, cannot contribute much to the iteration gain, since the delay spread is much shorter than the resolution of the channel.

To demonstrate that the orthogonal ISI cannot be used to improve the error rates, a transmission channel with only orthogonal taps is examined in the following. The channel impulse response h(t) is given by

$$h(t) = g(t)a(t) = (0.5j\delta(t+T) + \delta(t) - 0.5j\delta(t-T))a(t)\,. \qquad (4.1)$$

The impulse response, g(t), models the time-invariant channel impulse response of the modulator and the complex factor a(t) models a one-tap time-variant Rayleigh fading channel. This model roughly describes the transmission channel of GSM systems in which the delay spread is small.

In Fig. 4.7 the BER (class 1) after the 0th iteration and after the 1st iteration are depicted for scheme 1a. The Doppler frequency is selected to be 41.1 Hz, corresponding to a velocity of 50 km/h at a carrier frequency of 900 MHz. The coding and interleaving scheme TCH/FS is used. The modulator and the fading channel are modeled as described above. No iteration gains are obtained for this type of channel.

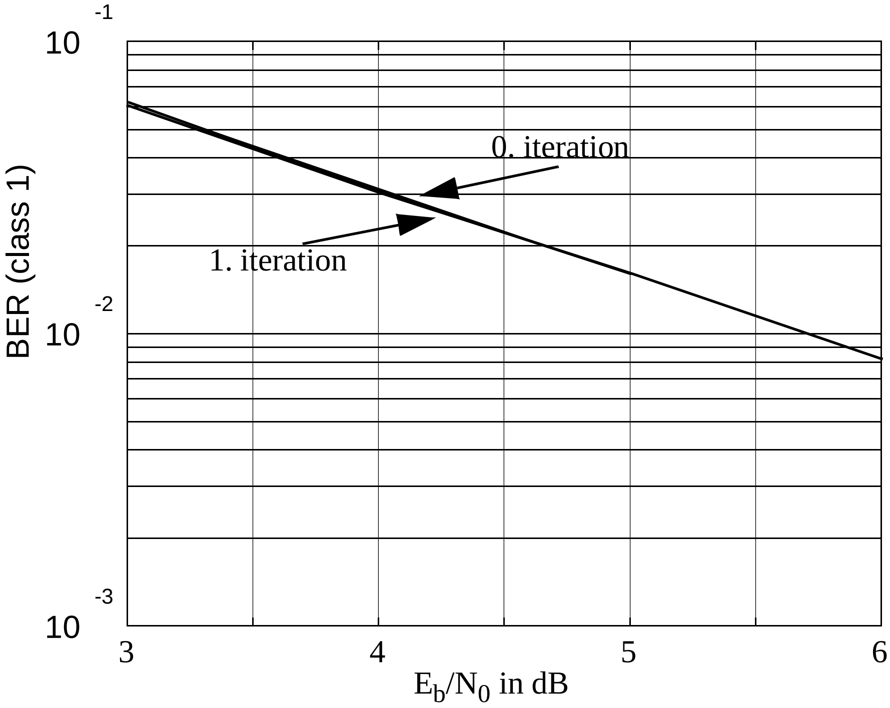


Figure 4.7. BER of class 1 bits for channel with time-variant component $g(t) = \frac{1}{2}j\delta(t+T) + \delta(t) - \frac{1}{2}j\delta(t-T)$ and one-tap Rayleigh fading (50 km/h and carrier frequency of 900 Mhz) and ideal FH.

The information passed from the equalizer to the decoder stays unchanged for all iterations. The independent estimates from the neighbouring symbols do not influence the a posteriori information of the symbol itself.

Although no iteration gains are obtained for the decoded bits, the raw BER is improved through iterative equalization and decoding as illustrated in Fig. 4.8. Here, the BER after the equalizer are given. At a BER of 8% an iteration gain of about 1.9 dB is achieved. The a posteriori values of the SISO equalizer are improved.

As explained in Chapter 3, the a posteriori values $L^E(\hat{x}_i)$ of the equalizer are composed of three values: the channel value, y_i, the extrinsic values, $L_e^E(\hat{x}_i)$, and the a priori values, $L(x_i)$. The channel information together with the extrinsic value form the information, $L_*^E(\hat{x}_i)$, that is passed to the decoder. This values stay unchanged for all iterations. Only the a priori information $L(x_i)$ that has been equal to zero on the 0th iteration has changed and, hence, the a posteriori information $L^E(\hat{x}_i)$ is improved. This explains the improvement of the raw BER.

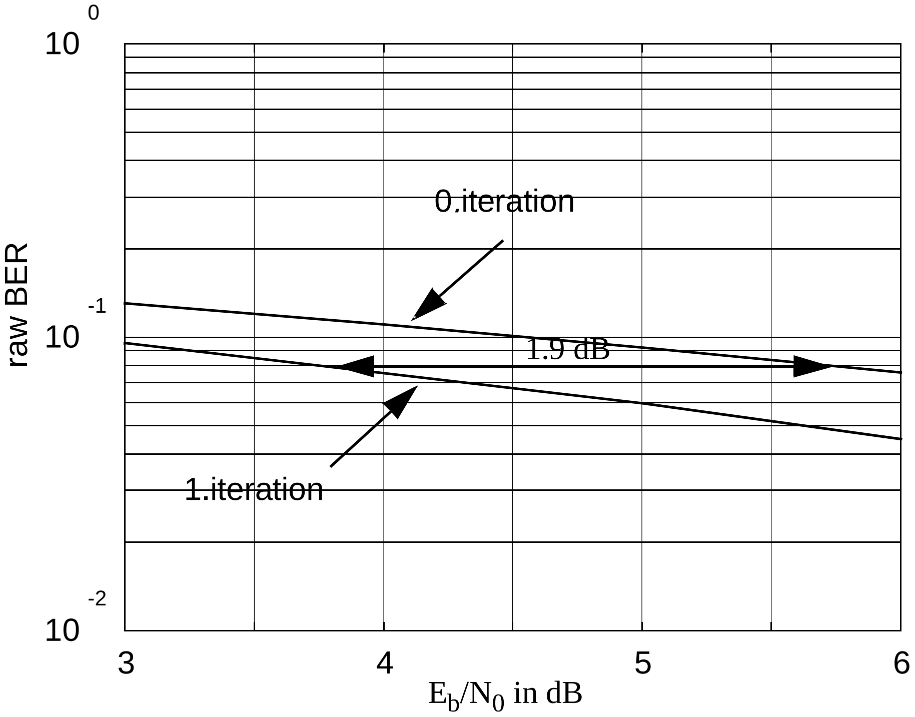


Figure 4.8. Raw BER for channel with time-variant component $g(t) = j0.5\delta(t+T) + \delta(t) - 0.5j\delta(t-T)$ and one-tap Rayleigh fading (50 km/h and carrier frequency of 900 Mhz) and ideal FH.

4.1.2 Delay-diversity for the full-rate speech traffic channel

Using the same modulator, larger iteration gains can be expected only if the delay spread of the radio channel is large. A scenario with large delay spreads is obtained if delay diversity is used for transmission [Win98]. In Fig. 4.9 delay diversity transmission in the downlink is illustrated. The BS (base station) transmits the data via two transmit antennas. The modulated signal s(t) is sent on transmit antenna a; and the delayed version s(t-T), on transmit antenna b. The transmitted signal is detected at the receive antenna of the MS (mobile station).

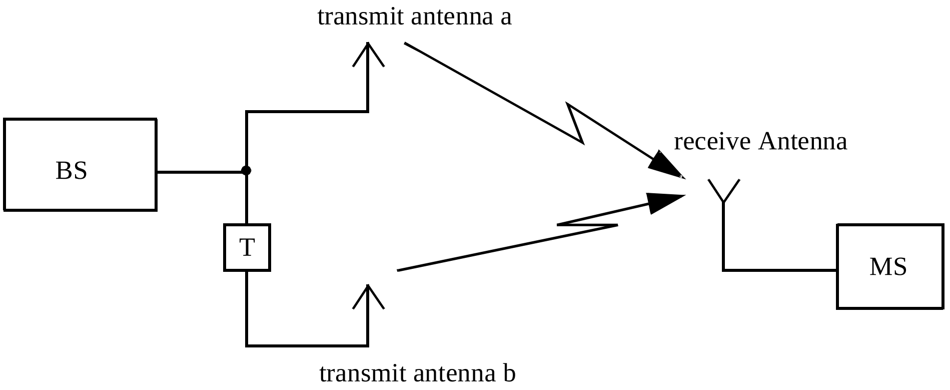


Figure 4.9. Transmission with delay diversity in the downlink.

If the delay spread of each transmission channel is negligible ($\tau_c \ll T$) and the distance of the transmit antennas is large enough, the compound radio channel observed at the receiver consists of two independent fading taps with relative time delay T (see Fig. 4.10).

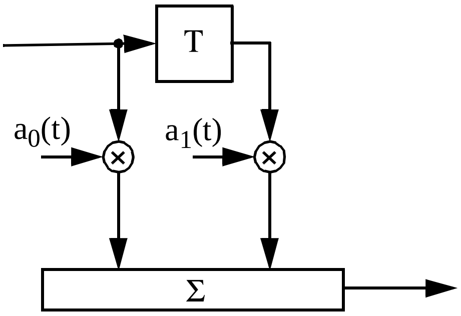


Figure 4.10. Tapped delay line model for the delay diversity mobile radio channel.

In Fig. 4.11 the FER for turbo-detection for a TU50 channel with ideal FH and delay diversity is depicted. Compared to conventional transmission, the delay diversity scheme performs about 0.7 dB better at a FER of 3%. Without delay diversity a maximal gain of 0.1-0.2 dB can be achieved (see Fig. 4.1). With delay diversity an iteration gain of about 0.4 dB can be obtained with scheme 2.

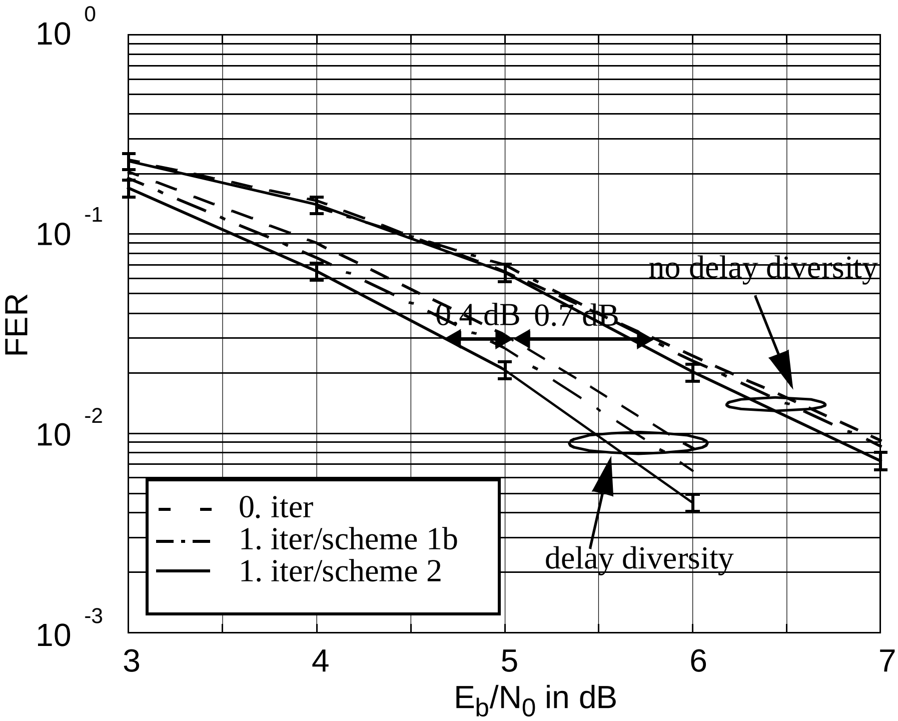


Figure 4.11. FER (TCH/FS) versus E_b/N_0 with and without delay diversity on TU50-channel with ideal FH.

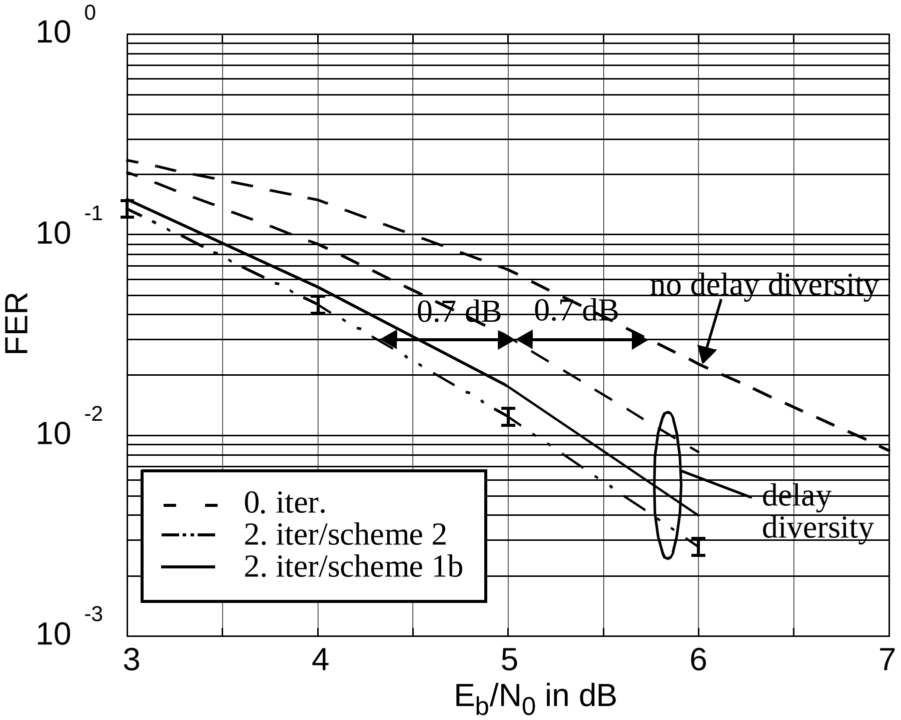


Figure 4.12. FER (TCH/FS) versus E_b/N_0 with and without delay diversity on TU50-channel with ideal FH forwarding the first correctly detected frame.

Again, it is possible to forward the data once the block has been detected correctly, see Fig. 4.2. As shown in Fig. 4.12, iteration gains of 0.5 dB and 0.7 dB are achieved with the schemes 1b and 2, respectively. Note that these iteration gains are already achieved on the first iteration.

However, for two reasons the gains do not approach the results of Chapter 3. First, the relative phase of the two independent fading taps is equally distributed. Since only the non-orthogonal part of the interference can be exploited in turbo-detection, the entire energy of the second tap does not contribute to the turbo-detector. Second, the delay spread is only visible to the detector if the amplitudes of both taps are in the same range. If one tap is strong and the other suffers a deep fade, the situation for the turbo-detector is again the same as without delay diversity.

4.1.3 The general packet radio service

GPRS is a new standard introduced within GSM “phase 2+” [Hae98]. In order to allow flexible data rate allocation for various data services, instead of utilizing circuit switched transmission as in HSCSD or TCH/F9.6 (traffic channel/full-rate 9.6 kbit/s), packet switched transmission is exploited to meet the needs of flexible data services. For instance, in internet browsing most data is transmitted in the downlink and not at a constant rate. In conventional circuit switched transmission the same bandwidth would be allocated for the uplink even if only small data rates are transmitted in that direction. Packet switching is also more capacity-efficient since the time slot is not allocated to the same subscriber for the entire duration of the connection. If no data has to be transmitted, a time slot can be used for other data services.

Four coding schemes, CS-1 to CS-4, see Table 4.1,which permute data rates from 9.2 kbit/s to 22 kbit/s per time slot, are specified in [ETS98b]. For higher data rates several slots can be bundled and used for the same service. With this technique data rates of more than 64 kbit/s can be provided.

Scheme	block size	blockcode	tail	conv.code R	puncturing	data rate in kbit/s
CS1	184	40	4	1/2	0	9.2
CS2	274	16	4	1/2	132	13.7
CS3	312	16	4	1/2	220	15.6
CS4	440	16	0	not applied	0	22

Table 4.1. The parameters of the four coding schemes of GPRS.

Depending on the coding scheme, a frame of 184 to 440 bits is first block encoded. Except for CS-4, the block encoded data is convolutional encoded and punctured. The resulting 456 code bits are intrablock interleaved. The interleaved bits are distributed over four bursts, modulated and sent. The same GMSK modulator as specified for the other transmission schemes in GSM is used.

Investigations of GMSK modulation show that iterative equalization and decoding only marginally improves the error performance of the decoder. Carrying out the same simulations as in Section 4.1.1 does not provide more insight since the same poor results are expected. In order to obtain gains, the turbo-principle has to be applied differently.

In GPRS a serially concatenated code system composed of the blockcode and the convolutional code is used. In the following, the turbo-principle is applied to iteratively decode this coding system, see Fig. 4.13. Note that here no interleaver is used between the two decoders. This serially concatenated code can also be decoded as a single code. However, the ML decoding of the entire code is too complex. The number of states would be 2^{20}. The iterative decoding of the constituent code approaches the ML solution with reduced complexity.

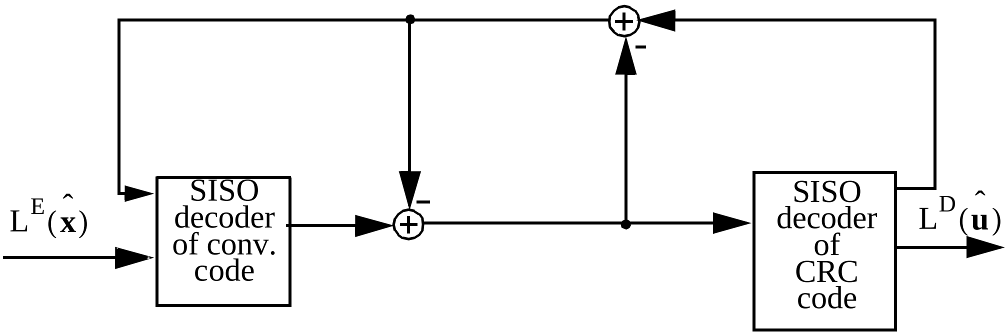


Figure 4.13. Block diagram of iterative decoding of the block code and the convolutional code.

Since the block code is originally used to detect erroneous transmitted frames at the receiver and to guarantee low residual block error rates, the examination has to be carried out very carefully. The same RFER (residual FER) has to be maintained. For iteratively decoding of the concatenated code system, first a new decoding strategy that enables also the detection of erroneous blocks is utilised.

The SISO algorithm of Section 2.2.2 is used to both, to correct errors more effectively than with conventional hard-decision decoding as well as to detect those frames that are still not correct, see Appendix D. If the L-values are large, it can be assumed that the decisions are correct. If the L-values are small, an error is very likely and the frame is rejected.

In the Appendix D, it was shown that the throughput of an ARQ scheme can be improved by SISO decoding without losing error detection capabilities. The probability of undetected errors is maintained. In order to guarantee the same RFER, good packets are discarded, and thus a large fraction of the SISO decoding gain is sacrificed. In order to obtain a significant improvement, the soft-decision gain has to be large. Otherwise, the remaining performance gain is negligible.

In the coding schemes, CS-2 to CS-4, a CRC code with 16 parity bits is used. The generator polynomial specified in [ETS98b] is given by

$$g(\quad) = 1 + \quad + \quad + D^{16}. \qquad (4.2)$$

In order to decode this code with the trellis-oriented SISO decoder given in Section 2.2.2, a trellis with n=2^{16}=65536 states is constructed. As the trellis of a CRC code is not regular, i.e. the trellis changes from stage to stage, the decoder needs to recalculate the trellis with 65536 states and 131072 transitions for each stage.

In order to reduce computational complexity and the memory needed to store the entire trellis, an equivalent non-systematically encoded code with the same generator polynomial as the CRC code is used, see Section 2.1.1.2. This code has a regular structure, i.e. the trellis is identical for all stages. For this regular trellis the same SISO decoder as for a convolutional code can be exploited.

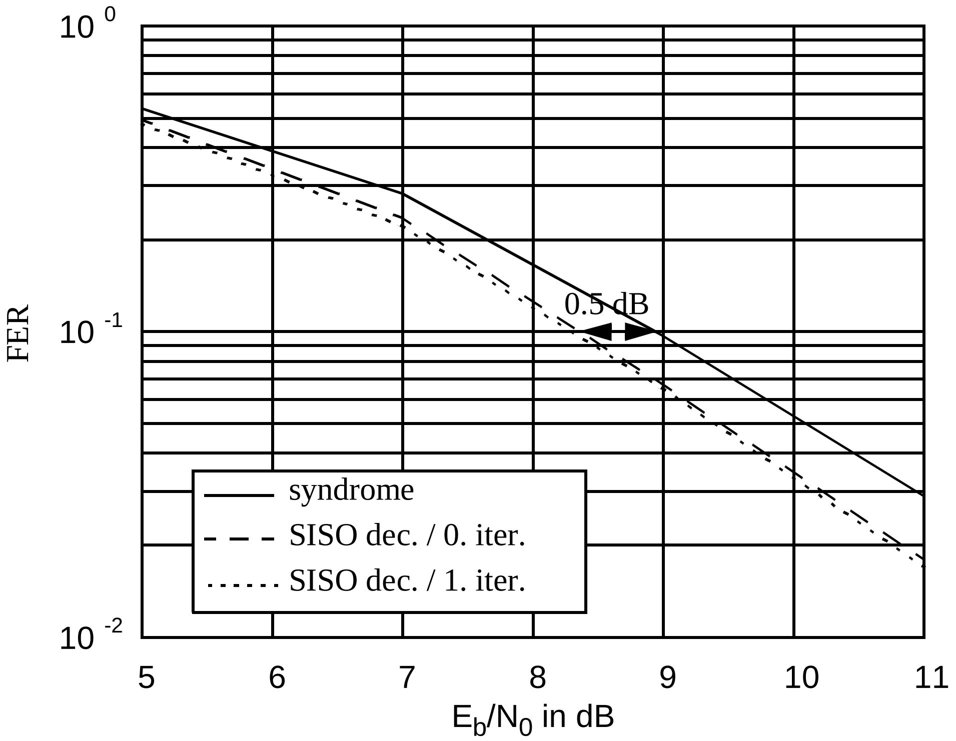


Figure 4.14. FER for 16 bit CRC code for TU3 with ideal FH and GPRS CS-2 parameters.

In Fig. 4.14 the gain for SISO decoding of the shortened cyclic block code is depicted. The coding and transmission scheme is chosen similar to that used in CS-2. A block of 274 bits is first encoded with the shortened non-systematically encoded cyclic code with the generator polynomial from (4.2). Four tail bits for the termination of the memory four convolutional code of rate R=1/2 are appended to the block-encoded 290 bits. The 588 convolutional encoded bits are then punctured, and the remaining 456 bits are block diagonal interleaved and mapped onto four normal bursts. The only difference from CS-2 is that the USF (uplink state flags) are not treated, and instead of the CRC-16 code an equivalent non-systematically encoded code is used. The radio channel is a TU3 channel with ideal FH. GMSK modulation with BT=0.3 is applied. By using a SISO decoder for the block code a gain of slightly more than 0.5 dB can be obtained. If the same RFER as for error detection has to be achieved, the unreliable blocks must be discard. Using the same criteria as in Appendix C, the gain obtained from SISO decoding vanishes.

The reason for the small difference between hard-decision decoding and SISO decoding of the CRC code is the error distribution after the convolutional channel decoder. Most of the frames passed to the block decoder are either correct or have more than ten errors. The SISO block decoder is not able to correct these severely distorted blocks. With a redundancy of only 16 bits only a few errors per frame can be corrected. The situation of Appendix C is different. Here, the transmission channel is AWGN and no convolutional coding is applied. Hence, about 50% of the frames have less than np errors where n is the blocklength, and p the bit error rate of the BSC (binary symmetric channel). For a bit error rate of 0.01 most of the blocks of length 920 bits have less than 3 errors, thus, a much higher improvement can be achieved.

In addition, Fig. 4.14 shows that only marginal gains are achieved by iterative decoding of the concatenated coding system. Since the outer SISO block decoder can only generate small incremental information, the inner convolutional decoder cannot profit from the a priori information supplied by the outer decoder. Another reason for the small gains is that there is no interleaver between the block coder and the convolutional coder. Hence, the values passed between the two decoding devices are strongly correlated.

In summary, SISO decoding can only improve the FER by about 0.5 dB. Iterative decoding lowers the FER marginally. In order to keep the properties of error detection, i.e. the low probability of undetected erroneous frames, the achieved gain diminishes. Therefore, SISO block decoding and iterative decoding are not appropriate methods to improve the performance of GPRS CS-2 to CS-4.

4.2 Turbo-detection for enhanced data services in GSM

At the time of the investigations, new data services were being standardized by the ETSI. The new data services are called EDGE [ETS99]. In EDGE, similar to conventional GSM, two types of data services are specified: circuit and packet switched services. The packet switched service is called EGPRS; and the circuit switched service, ECSD. EGPRS (enhanced GPRS) is a evolution of the currently available GPRS standard, while ECSD is the successor to HSCSD.

Not all parameters for coding, interleaving, and burst formatting are settled. In our work the parameters described in [ETS98e] and [ETS98f] are used. It is not claimed that all parameters of these documents will be adopted for the standard. However, as the changes mainly affect puncturing patterns, generator polynomials, tail bits, etc., the main principles can be examined. The principle transmission scheme for EDGE is depicted in Fig. 4.15.

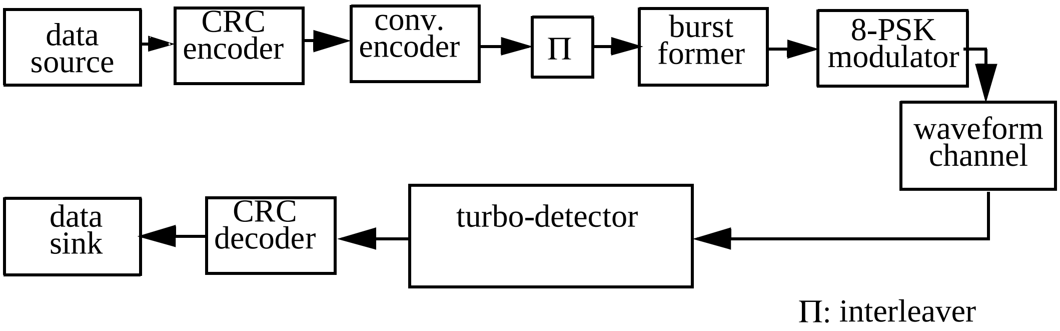


Figure 4.15. Principle transmission scheme for EDGE using turbo-detection.

The main purpose of EDGE is to provide higher data rates than in conventional GSM data services. Therefore, a more bandwidth efficient modulation technique is exploited: 8-PSK modulation. This higher-order modulation technique permits the transmission of three bits per symbol. In order to apply turbo-detection to EDGE services the equalizer must be modified. These modifications are valid for all higher-order M-ary modulation techniques.

4.2.1 General principles for higher-order M-ary modulation

With higher-order M-ary modulation more than one bit can be transmitted with a modulated symbol. The number of bits per symbol is equal to $\lfloor \mathrm{ld}(M) \rfloor$, where M is the alphabet size of the modulation. The SISO equalizer has to be changed to be able to detect the M-ary symbols and generate soft-output information about the bits. For equalization of M-ary symbols the following characteristics have to be taken into account:

- Trellis complexity: The number of states is equal to M^L with L being the memory length of the channel. If L is equal to four, the number of states is 16 for binary modulation, whereas for octary modulation the number of states is 4096. The number of transitions per state is M. The number of transitions per trellis stage is M^{L+1}. If L is equal to four for binary and octary modulation, there are 32 and 32768 transitions, respectively. Thus, higher-order modulation requires a significantly higher complexity with increasing channel length. For trellis-oriented equalizers this means that the complexity is already high for relative short channels.
- A priori values for the equalizer: In turbo-detection the a priori values for the equalizer are generated by the decoder. The decoder generates soft-output, $L_e^D(\hat{\mathbf{x}})$, for the code bits. In order to detect the symbols the equalizer has to transform these L-values for the bits into L-values of the symbols.
- Soft-output of the equalizer: The SISO equalizer has to generate L-values about the bits. Hence, the calculation of the soft-output of the equalizer has to be modified.

The decoder passes L-values, $L(x_{i,j})$, of the single bits $x_{i,j}$ to the equalizer. Here, $x_{i,j}$ represents the j^{th} bit of the i^{th} symbol d_i. As mentioned above, these L-values have to be converted to L-values corresponding to symbols. Similar to the binary case the log-likelihood ratios are compared to a reference symbol d_r. For each symbol d of the symbol alphabet $D = \{0, 1, \ldots, M-1\}$ the symbol L-values $L_d(d_i)$ have to be calculated. The symbol L-value $L_d(d_i)$ is defined by

$$L_d(d_i) = \ln\frac{P(d_i = d)}{P(d_i = d_r)} = \ln\frac{\prod_{j=1} P(x_{i,j} = d(j))}{\prod_{j=1}^{ld(M)} P(x_{i,j} = d_r(j))}, \tag{4.3}$$

with d(j) being the j^{th} bit of the tuple corresponding to symbol d, while $d_r(j)$ is the j^{th} bits of the tuple corresponding to the reference symbol d_r. It is convenient to take the symbol corresponding to the $\lfloor ld(M) \rfloor$-tuple of ones as the reference symbol d_r. Thus, the relation between the symbol L-values $L_d(d_i)$ and the bit L-values $L(x_{i,j})$ is given by

$$L_d(d_i) = \sum_{j=1, (d(j)=0)}^{ld(M)} L(x_{i,j}). \tag{4.4}$$

The normalized logarithm, $\Gamma_{i,d}(s',s)$, of the transition probabilities corresponding to the d-transmission can then be calculated by

$$\Gamma_{i,d}(s', s) = \log\{\gamma_{i,d}(s', s)\} = L_d(d_i) - \frac{|y_i - c_{s',d}|^2}{2\hat{\sigma}_N^2}, \tag{4.5}$$

with $c_{s',d}$ being the reference value corresponding to the d-transmission originating in state s’.

With the MAP algorithm the soft-values of the bits can be calculated by

$$L(\hat{x}_{i,j}) = \log \frac{\sum\limits_{(s',s),\, x_{i,j}=0} P(S_i = s', S_{i+1} = s, \mathbf{y}, L(\mathbf{d}))}{\sum\limits_{(s',s),\, x_{i,j}=1} P(S_i = s', S_{i+1} = s, \mathbf{y}, L(\mathbf{d}))} . \qquad (4.6)$$

Here, for each stage and each of the ld(M) bits all branches that correspond to the same bit are summed.

4.2.2 Turbo-detection for enhanced general packet radio service

In EGPRS various coding and modulation schemes will be specified [ETS99]. The schemes using 8-PSK modulation will achieve data rates from 22.8 kbit/s to 69.2 kbit/s per time slot, whereas in conventional GPRS only data rates up to 21.55 kbit/s can be transmitted per time slot. Depending on the quality of the radio channel, the transmission alternates between the various schemes. This environment-dependent variation of the coding and modulation scheme is called link adaptation. If the channel is poor, the modulation is changed from 8-PSK to the more robust GMSK; again the coding schemes of GPRS are used. From Section 4.1 it is known that turbo-detection for GMSK modulated signals in GSM does achieve significant iteration gains. Hence, only the services using 8-PSK modulation are considered in the following.

In EDGE all data blocks are first block encoded for the purpose of error detection. Since the parameters of the block code have not been settled at the time of this work and since it is not necessary to investigate the behaviour of the block codes in order to evaluate the potential of turbo-detection, block coding is not considered in the following. In the simulations ideal error detection is assumed, so that the RBER is equal to zero.

Depending on the coding scheme, a block of 456 to 1384 bits[1] is convolutionally encoded and punctured. The resulting 1384 bits are then block rectangular interleaved and four bursts are formed. Note that intrablock interleaving is used. The bursts are 8-PSK modulated and transmitted via the mobile radio channel.

First, the preliminary coding scheme PCS-2 [ETS98e] is considered. Six tail bits for the termination of the convolutional coder are appended to a block of 686 information

[1] These bits include six tail bits for the termination of the memory six convolutional code.

bits. These 692 bits are convolutionally encoded with the generator matrix **G**(D) given by

$$\mathbf{G}(\mathrm{D}) = (1 + \mathrm{D}^2 + \mathrm{D}^3 + \mathrm{D}^5 + \mathrm{D}^6, 1 + \mathrm{D} + \mathrm{D}^2 + \mathrm{D}^3 + \mathrm{D}^6)\,. \tag{4.7}$$

The resulting 1384 bits are then block rectangular interleaved. For each of the four bursts two stealing flags are inserted. The 1392 bits are then mapped to 464 8-PSK-symbols. Similar to GSM, 116 symbols are transmitted within a normal burst. The burst tail bits and the midamble bits are mapped to “+1”-symbols and “-1”-symbols. Therefore, the same channel parameter estimators as in GSM can be employed.

The equalizer used in the following simulations for 8-PSK modulation has $512 = 8^3$ states.

The FER after the decoder is given versus E_b/N_0 per modulated bit for a TU3-channel with ideal FH in Fig. 4.16. The equalizer is based on the Max-Log-MAP algorithm. At a FER of 10%, a gain of 2.1 dB can be achieved with four iterations. The largest gain of 1.5 dB is obtained on the first iteration whereas the second iteration only improves the FER by 0.4 dB. The final two iterations bring only small iteration gains.

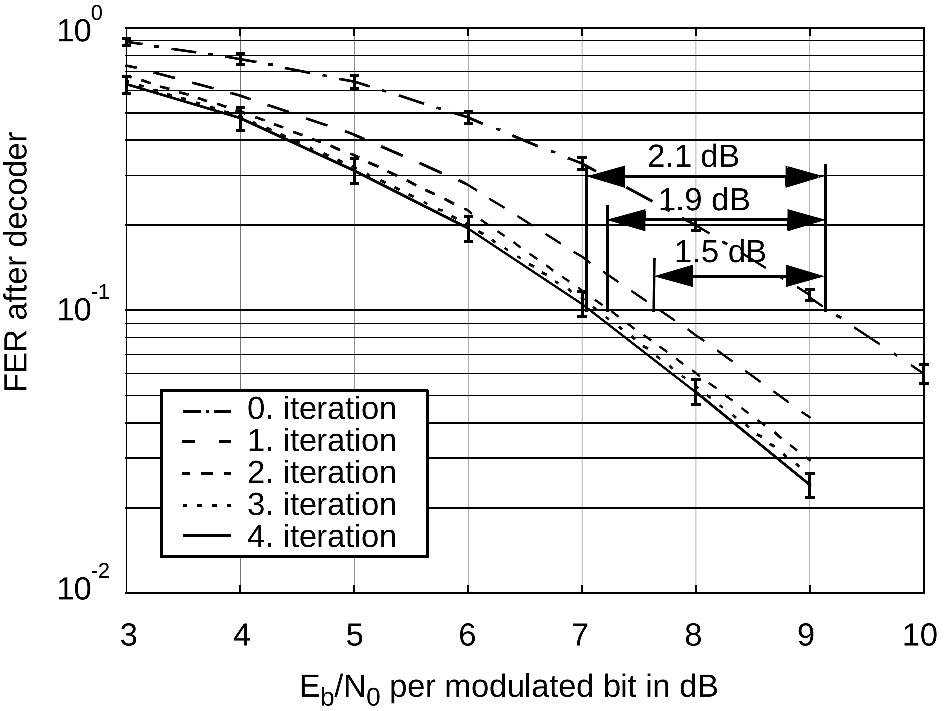


Figure 4.16. FER after the decoder versus E_b/N_0 per modulated bit for PCS-2 on a TU3-channel with ideal FH.

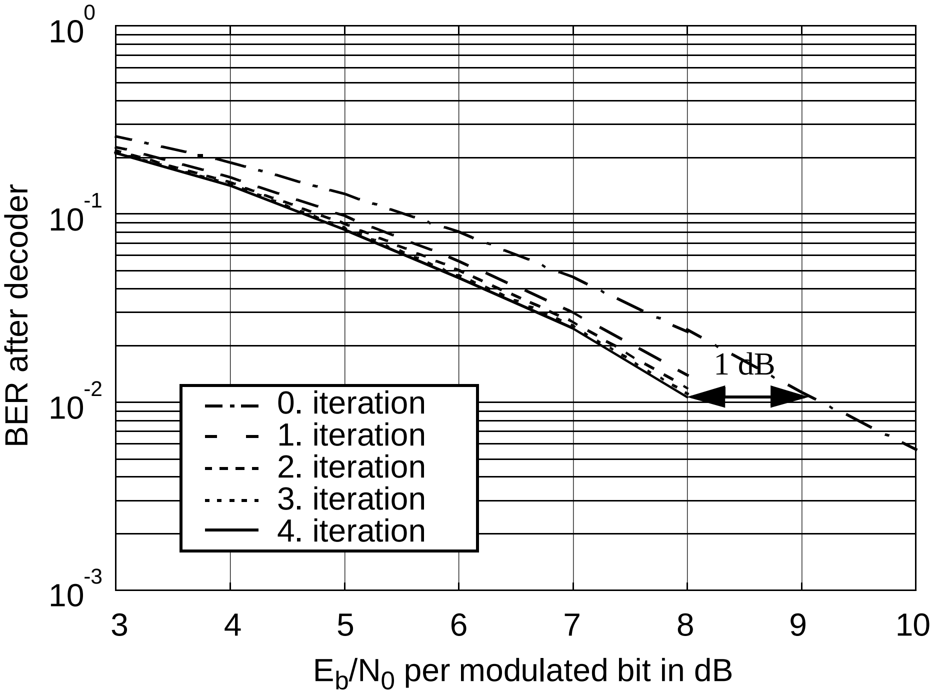


Figure 4.17. BER after the decoder versus E_b/N_0 per modulated bit for PSC-2 on a TU3-channel with ideal FH.

As illustrated in Fig. 4.17, the iteration gains for the BER after the decoder are not as high as those of the FER; a gain of only 1 dB is obtained due to the fading environment. On fading channels some blocks experience many errors. The bit errors in these bad blocks cannot be erased, and hence, the BER is determined by these erroneous blocks.

In Fig. 4.18, the iteration gains for the raw BER after the equalizer are given. The gain is about 5 dB at a BER of 8%. Due to the code constraints and the frequency diversity, the decoder can strongly improve the raw BER.

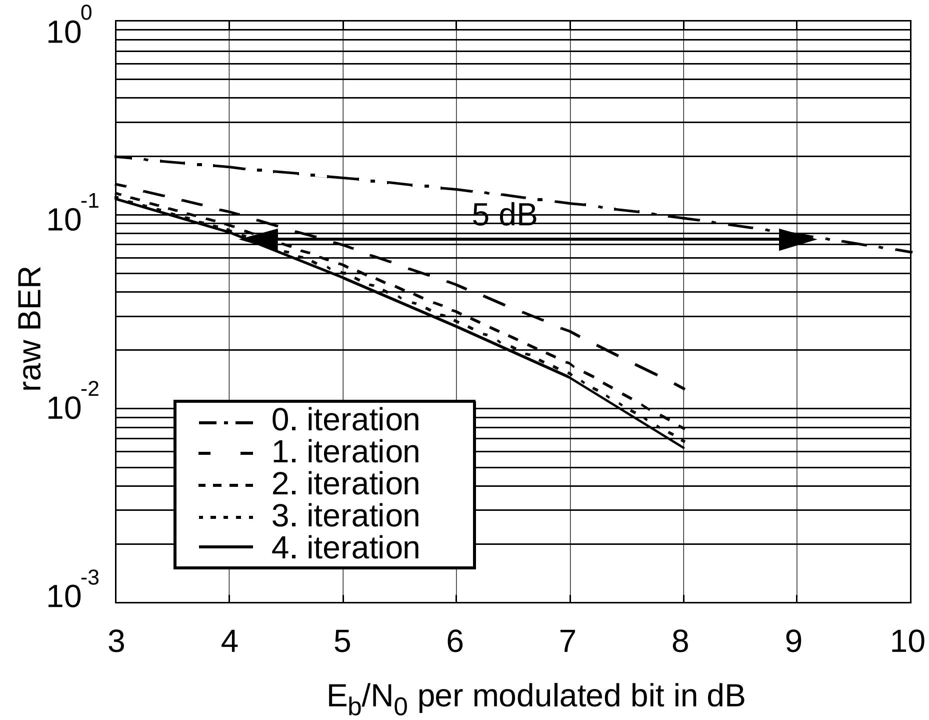


Figure 4.18. Raw BER after the equalizer versus E_b/N_0 per modulated bit for PSC-2 on a TU3-channel with ideal FH.

In Fig. 4.16, Fig. 4.17, and Fig. 4.18 a Max-Log-MAP equalizer is used. As already mentioned, the Max-Log-MAP algorithm is suboptimum. In Fig. 4.19 the FER using a Log-MAP equalizer and a Max-Log-MAP equalizer are compared after the 0th and the 4th iteration. After the 4th iteration the FER performance of the Log-MAP based scheme is only 0.1 dB better than the performance of the Max-Log-MAP based scheme.

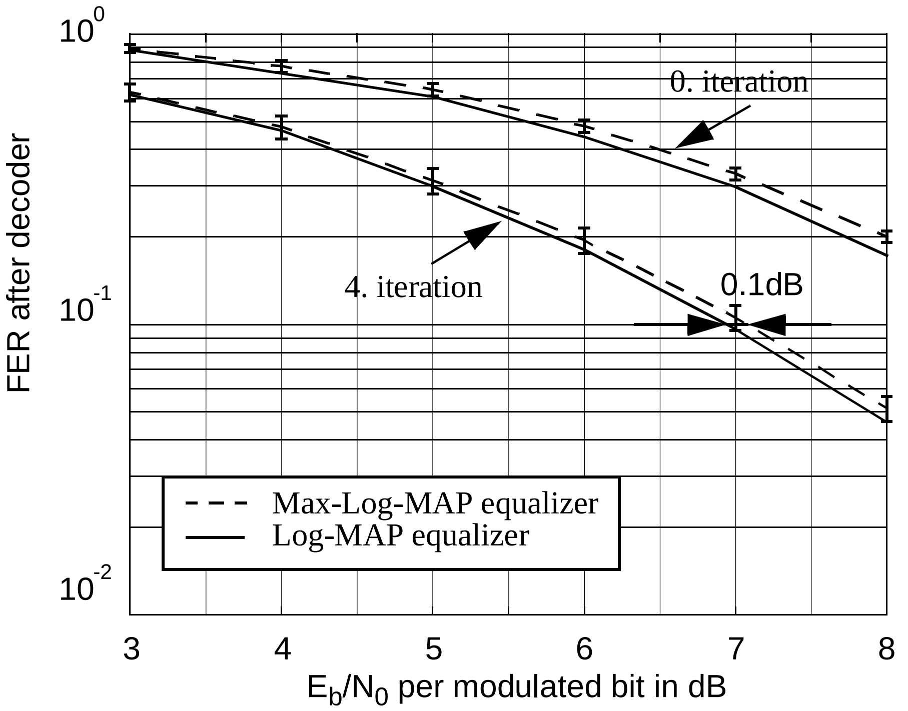


Figure 4.19. FER after the decoder versus E_b/N_0 per modulated bit for PCS-2 on a TU3-channel with ideal FH using a Log-MAP equalizer and a Max-Log-MAP equalizer.

In EGPRS the forward correction coding scheme is combined with an ARQ protocol. A measure for the capacity of ARQ schemes is the throughput. Assuming a selective-repeat ARQ protocol, with R_b being the data bit rate in kbit/s, the throughput ρ can be calculated by

$$\rho = (1-\mathrm{FER})R_b \tag{4.8}$$

if the frame erasures are uncorrelated [AnM91, LiC83]. The assumption of uncorrelated frame errors holds for ideal FH as consecutive frames are not interleaved.

On the basis of the results from Fig. 4.18, the throughput ρ for PCS-2 is depicted for several iterations in Fig. 4.20. The throughput can be improved by about 8 kbit/s at an E_b/N_0 of 7 dB.

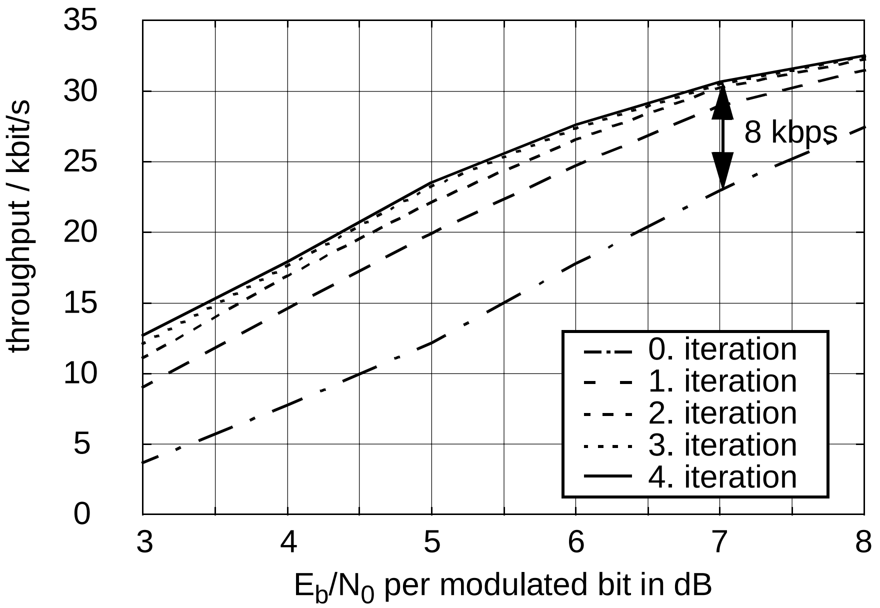


Figure 4.20. Throughput versus E_b/N_0 per modulated bit for PSC-2 on a TU3-channel with ideal FH for a selective-repeat ARQ protocol.

Important environments for GSM systems are interference limited areas such as city centers and pedestrian precincts. To evaluate the benefit of turbo-detection in these areas, the C/I-(c̲arrier to co-channel i̲nterferer)-performance has to be examined. Fig. 4.21 shows the FER after the decoder versus C/I for the TU3 channel with ideal FH. After the 4th iteration a gain of about 1dB is achieved. The largest fraction of this gain, i.e. 0.8 dB, is already obtained during the first iteration. Compared to the receiver sensitivity performance, the gain is half. A reason for this is that the equalizer falsely assumes white Gaussian noise. The interferer noise is strongly correlated, thus the channel representation in the receiver is inaccurate. Consequently, the soft-values are distorted.

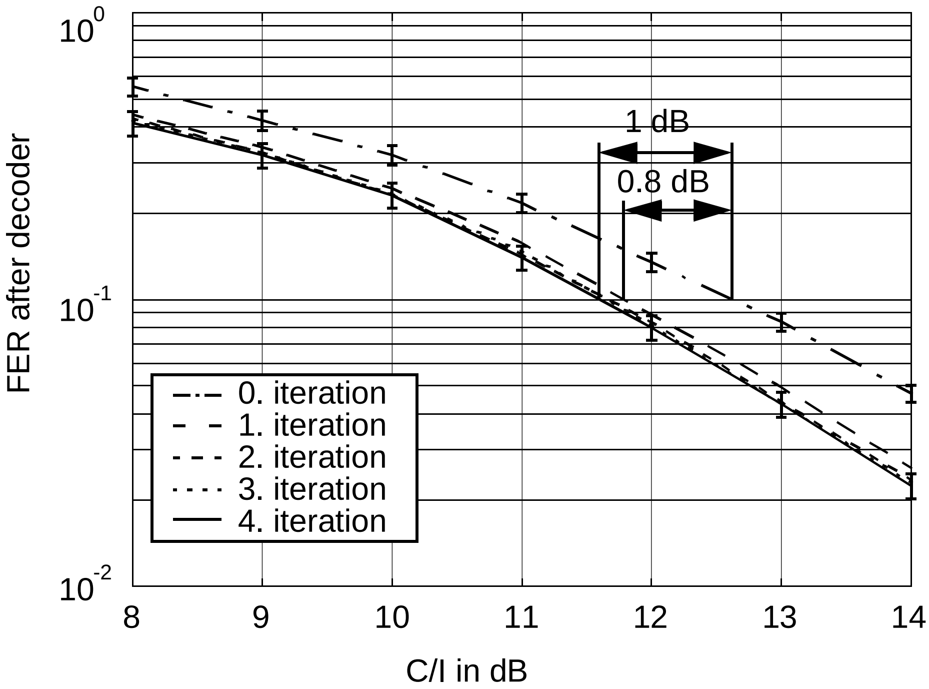


Figure 4.21. FER after the decoder versus C/I for PCS-2 on a TU3-channel with ideal FH.

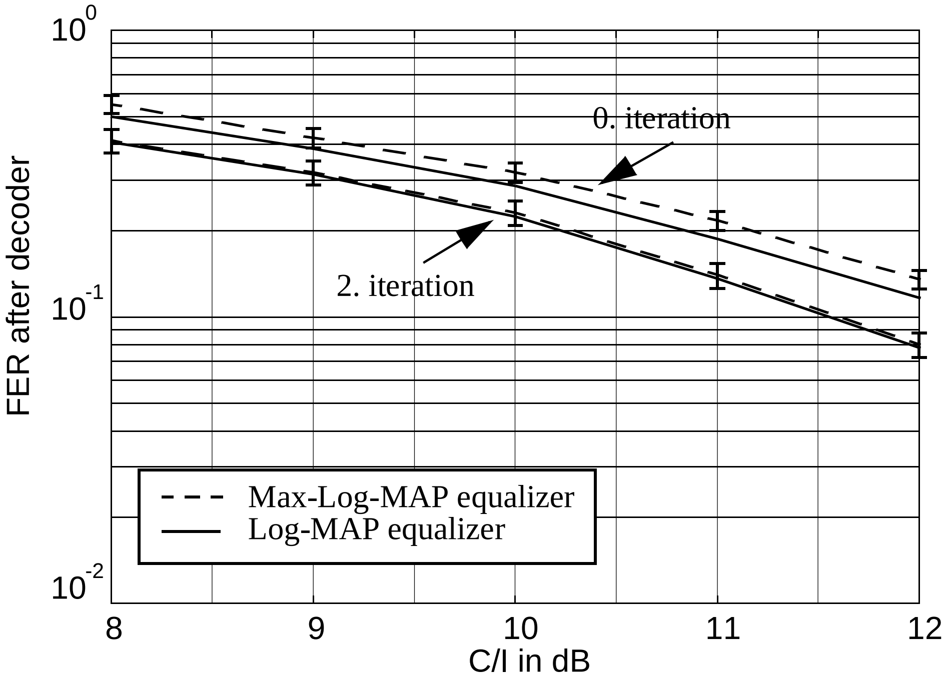


Figure 4.22. FER after the decoder versus C/I for PCS-2 on a TU3-channel with ideal FH using a Log-MAP equalizer and a Max-Log-MAP equalizer.

Comparing the performance of a Log-MAP and a Max-Log-MAP equalizer, the results of Fig. 4.22 are obtained. On the 0^{th} iteration the FER of the scheme using the Log-MAP algorithm outperforms the other scheme. However, after the 2^{nd} iteration the FER performance of both schemes is about the same. Since the complexity of the Log-MAP algorithm is larger than that of the Max-Log-MAP algorithm and since the FER performance is the nearly the same for both algorithms, see Fig. 4.19 and Fig. 4.22, the equalizer used in the following examinations for EDGE is based on the Max-Log-MAP algorithm.

In order to evaluate the applicability of turbo-detection to EDGE, several scenarios will be examined. In the following two different environments are treated:

- The TU 3 (no FH) channel: no frequency diversity is exploited at the decoder.
- The RA250 (no FH) channel: if non-adaptive equalization is used, the equalizer decisions become unreliable and the system performance suffers.

As mentioned above, the error performance of the system is worse if no FH is applied because the fading imposed on consecutive bursts is correlated, especially for low velocities of the mobile. A critical channel is the TU3 channel without FH, i.e. TU3 (no FH). In Fig. 4.23, the raw BER for a TU3 (no FH) channel are given. Compared to ideal FH, the gain of 5 dB in the ideal FH case shrinks to 3 dB. The reason is that the decoder cannot exploit the frequency diversity. Therefore, the turbo-component, not including the diversity information, is not able to improve the raw BER to the same extent.

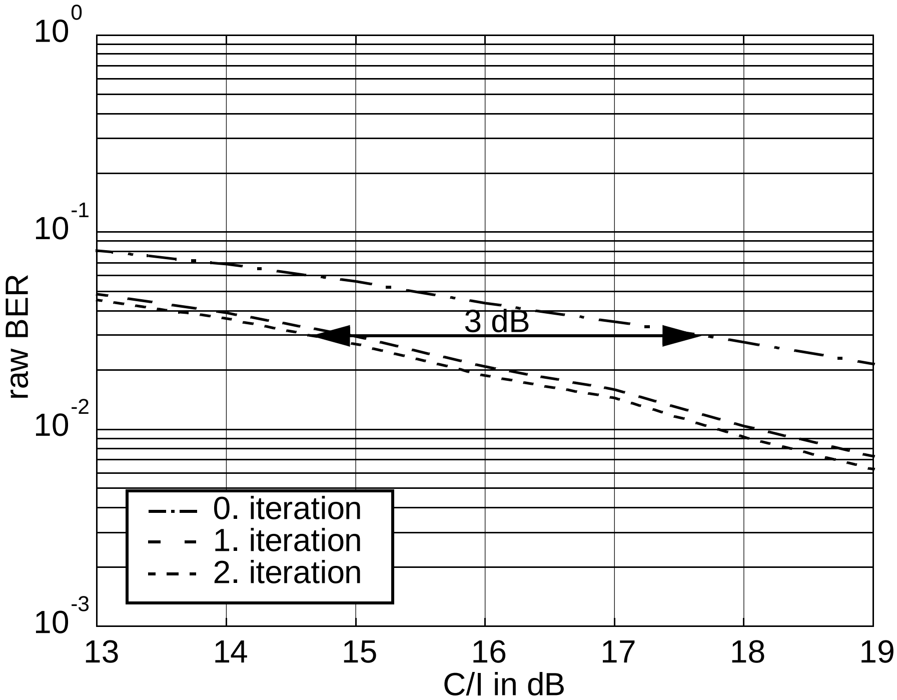


Figure 4.23. Raw BER versus C/I for PCS-2 on a TU3-channel without FH.

After the decoder even slightly larger iteration gains are achieved, as can be seen in Fig. 4.24. The turbo-detection gains are 1.0 dB and 1.3 dB after the 1st and 2nd iterations, respectively. In the case of FH, the diversity is fed back to the equalizer and improves the a posteriori information resulting in the large turbo-detection gains for the raw BER. However, the extrinsic information is not influenced by this diversity gain. The gains achieved after the decoder are in the same range. In order to achieve the same error performance after the decoder, the raw BER has to be better than if no FH is applied; the requires raw BER is 3% instead of 8%. Consequently, the soft-output values of the equalizer are more accurate gain in the latter case, enabling a higher iteration.

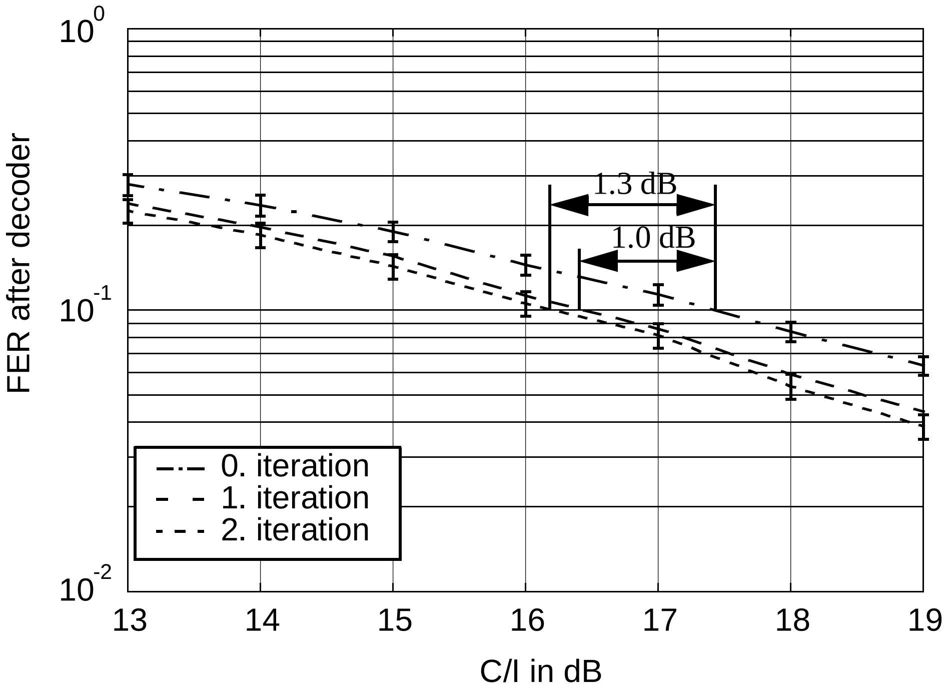


Figure 4.24. FER after the decoder versus C/I for PCS-2 on a TU3-channel without FH.

During non-adaptive channel estimation the channel impulse response is estimated from the midamble. If the mobile is moving rapidly, the channel impulse responses change significantly during a burst, and the channel impulse responses at the beginning and the end of the burst diverge strongly from the estimated responses. As will be described in Chapter 5, the soft-output of the equalizer suffers and the error performance of the system degrades.

In the following the impact of mismatched equalization on turbo-detection for EDGE is examined for the RA250 (no FH) channel. In Fig. 4.25 the FER are given for several iterations. The gain of about 2 dB on the TU3-channels can no longer be achieved; however, a gain of 1.2 dB is still obtained during the first iteration. Further iterations bring negligible gains.

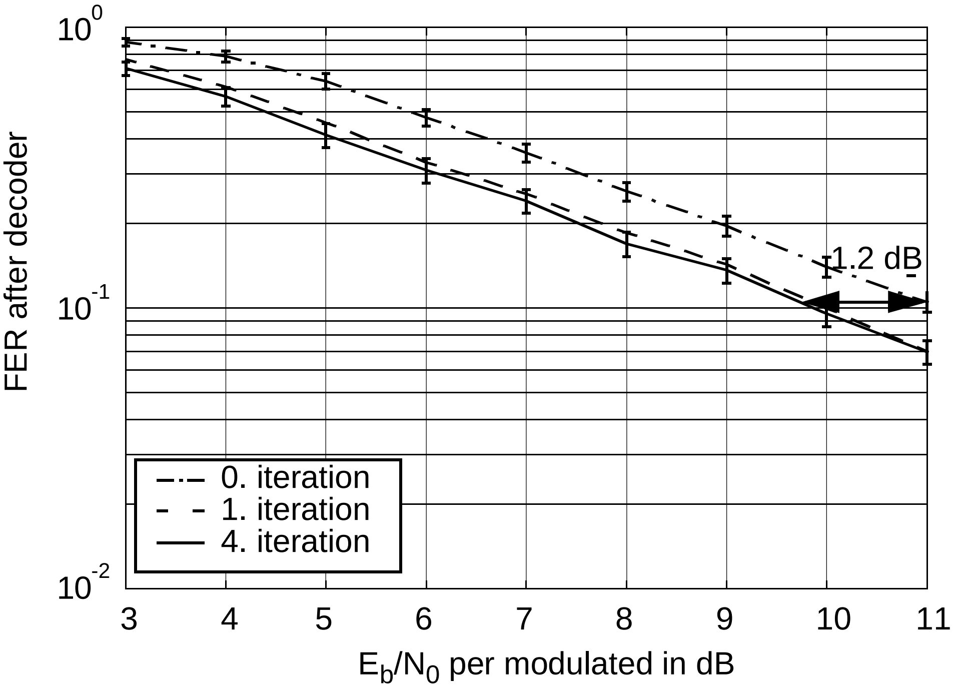


Figure 4.25. FER after the decoder versus E_b/N_0 per modulated bit for PCS-2 on a RA250 (no FH).

In the previous simulations it was shown that turbo-detection can improve the FER and, hence, the throughput of coding scheme PCS-2. However, in order to evaluate the benefits of turbo-detection for EDGE services it is not sufficient to investigate the performance of one coding scheme alone. For instance, from Fig. 4.20 it can be observed that a throughput gain of 8 kbit/s is obtained for PCS-2 at E_b/N_0=7 dB. However, this throughput gain affects EDGE only if PCS-2 is actually used at this specific E_b/N_0. It is also possible that the coding scheme PCS-3 already enables a higher throughput. Therefore, all coding schemes have to be considered together.

In [ETS98e] six different coding schemes, PCS-1 to PCS-6, with various code rates using 8-PSK modulation are presented. From simulation with these six coding schemes the FER versus E_b/N_0 and C/I are obtained. These results can be mapped on the throughput performance using (4.8). In Fig. 4.26 the throughputs versus E_b/N_0 per modulated bit after the 0th and the 4th iterations are depicted for the coding schemes

PCS-1 to PCS-6. For the coding schemes PCS-1 to PCS-5 iteration gains are obtained. For PCS-6 no iteration gains are obtained because no channel coding is applied.

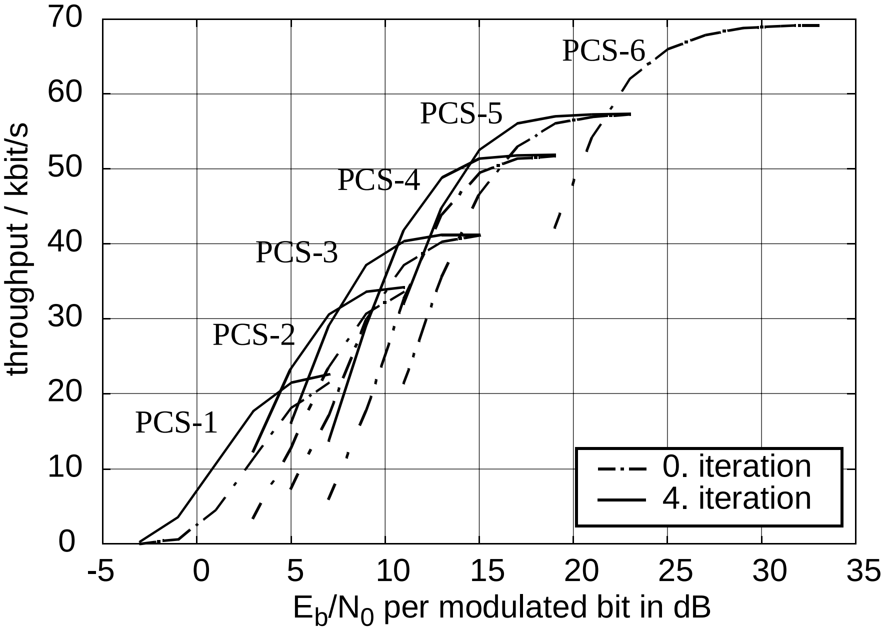


Figure 4.26. Throughput versus E_b/N_0 per modulated bit for PSC-1 to PCS-6 after the 0^{th} and the 4^{th} iteration on a TU3-channel with ideal FH for a selective-repeat ARQ protocol.

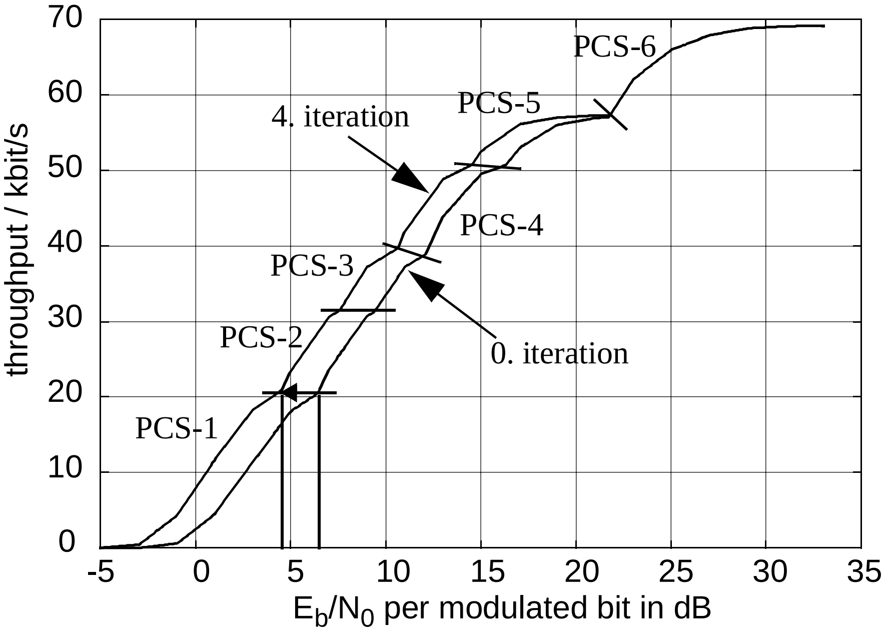


Figure 4.27. Maximum throughputs versus E_b/N_0 per modulated bit for PSC-1 to PCS-6 after the 0^{th} and 4^{th} iterations on a TU3-channel with ideal FH for a selective-repeat ARQ protocol and ideal link adaptation.

Assuming ideal link adaptation, data is always transmitted using the coding scheme with the highest throughput. Fig. 4.27 illustrates this principle. The throughputs are given for the best coding scheme after the 0th and 4th iterations. The area between the two curves denotes the turbo-detection throughput gain. It is also illustrated which coding scheme is used for which range of E_b/N_0. Note that if turbo-detection is employed, higher rate coding schemes are already used at lower E_b/N_0.

In Fig. 4.28 the throughput gain of turbo-detection after the 4th iteration compared to the 0th iteration is given. The local minima denote the points where the coding schemes are changed if turbo-detection is applied. The local maxima represent the points where the coding schemes are changed if conventional detection is applied.

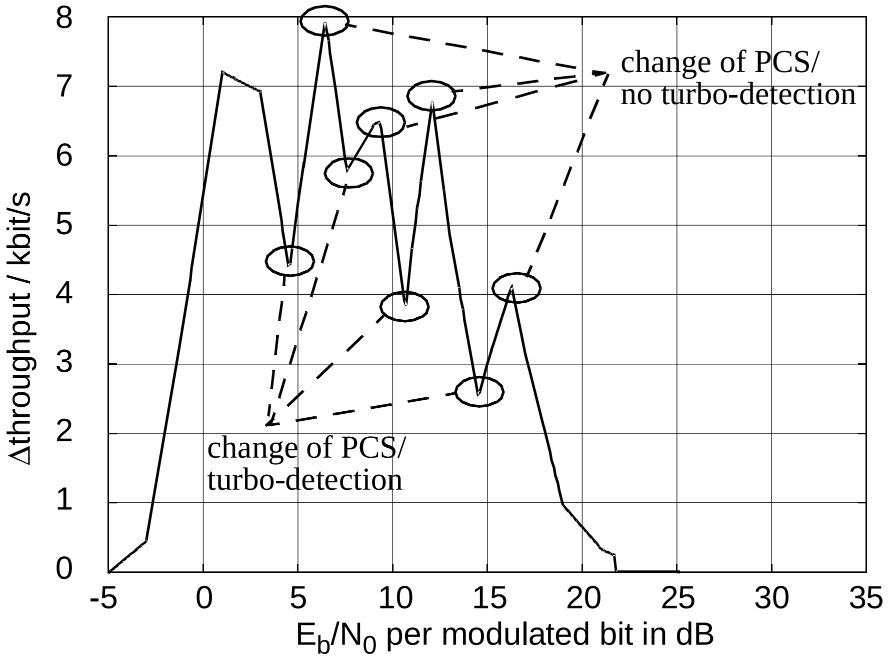


Figure 4.28.Throughput gain versus E_b/N_0 per modulated bit for PSC-1 to PCS-6 after the 0th and the 4th iteration on a TU3-channel with ideal FH for a selective-repeat ARQ protocol and ideal link adaptation.

It can be seen that the throughput gain depends on the E_b/N_0. Therefore, the E_b/N_0-distribution in the covered cell has to be known in order to get the gain in spectral efficiency. The distribution of E_b/N_0 depends on the environment as well as on parameters such as system load and power control. For simplicity a Gaussian distribution can be assumed.

A consistent observation in real mobile radio channels is that the deviations of the local mean value γ of the attenuation are lognormally distributed [Lee93]. This means that the logarithm of γ has a normal distribution:

$$p(\quad) = \frac{1}{\sigma_{dB}\sqrt{2\pi}} e^{\frac{-(\qquad)^2}{2\sigma_{dB}^2}}, \tag{4.9}$$

with γ_{db}, m_{dB}, and σ_{dB} being the local mean value of the attenuation, the mean, and the standard deviation, respectively, in dB. The values m_{dB}, and σ_{dB} depend on the environment and on other factors.

Example 4.1 *In Fig. 4.29 the cdf of γ_{db} per modulated bit is given for 8-PSK modulation and for GSMK modulation. This distribution corresponds to a 90% speech coverage for GMSK modulation, assuming a required E_b/N_0 of 6 dB. The standard deviation σ_{dB} is 6 dB. Since three bits are transmitted with one modulation symbol for 8-PSK modulation, the distribution for the GMSK modulation is shifted by 4.8 dB to get the distribution for 8-PSK modulation.*

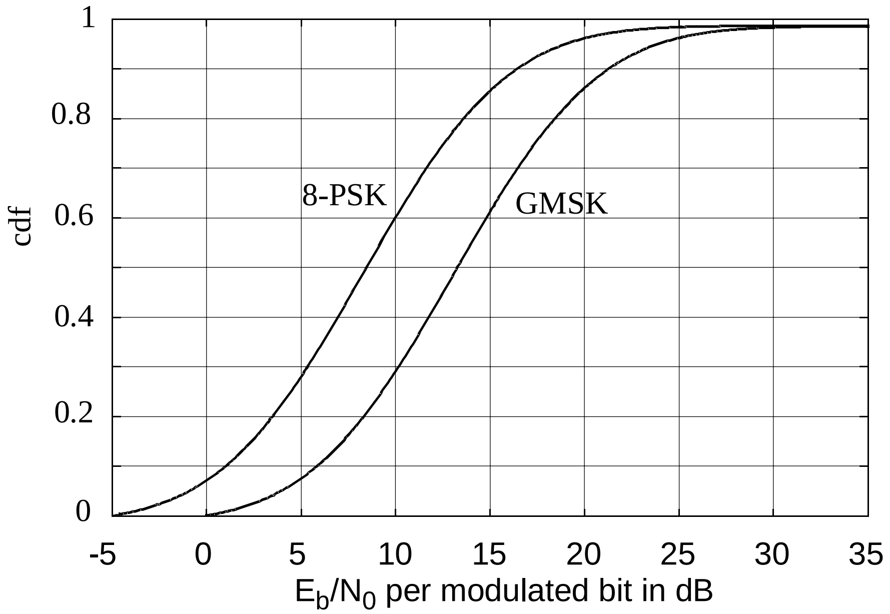


Figure 4.29. Cdf of local mean of E_b/N_0 per modulated bit for 8-PSK modulation and for GMSK modulation; 90% speech coverage for GMSK modulated TCH/FS; σ_{dB}=6dB.

In Fig. 4.30 and Fig. 4.31 the throughput gain and the relative throughput gain for various m_{dB} and σ_{dB} are depicted, respectively. For typical values the throughput gain ranges from 3 kbit/s to more than 6 kbit/s per time slot. If all time slots are allocated for EDGE services a throughput gain per carrier of 24 kbit/s/200kHz to 48 kbit/s/200kHz can be achieved.

The relative throughput gain Δg_r for various m_{dB} and σ_{dB} is defined by

$$\Delta g_r = \frac{\Delta \mathrm{throughput}}{\mathrm{throughput}|_{\mathrm{it0}}}. \qquad (4.10)$$

It is the absolute throughput gain divided by the absolute throughput at the 0^{th} iteration. It can be seen that the relative throughput ranges from 10% to 35%. The absolute and the relative gain for the parameters of Example 4.1 are indicated with black points. The throughput gain is 4.33 kbit/s and the relative gain is 11.5%.

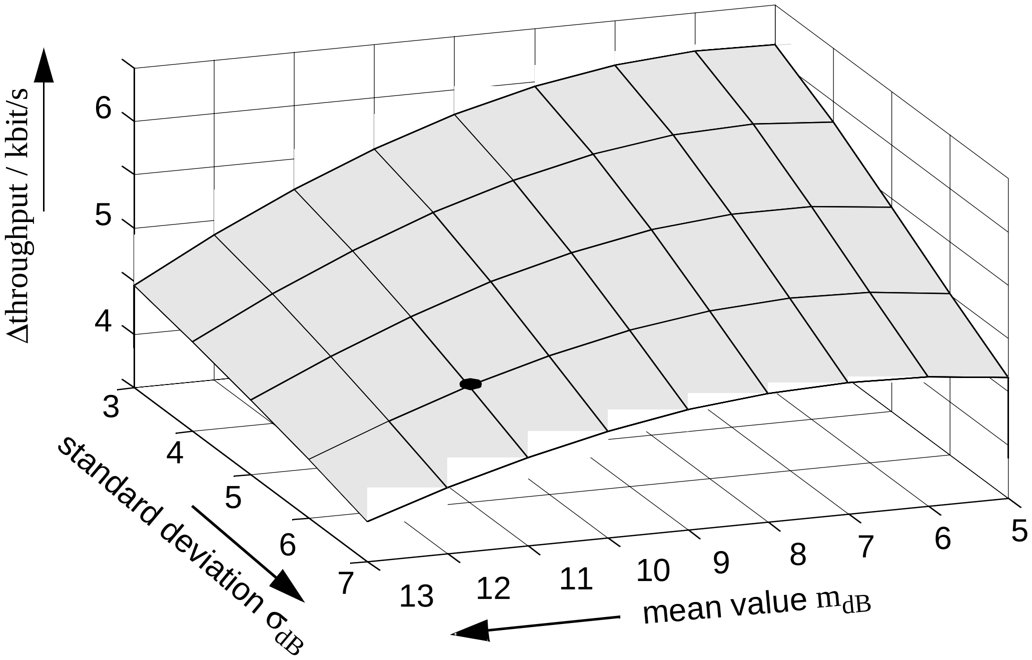


Figure 4.30. Throughput gain for selected m_{dB} and σ_{dB} on a TU3-channel with ideal FH for a selective-repeat ARQ protocol.

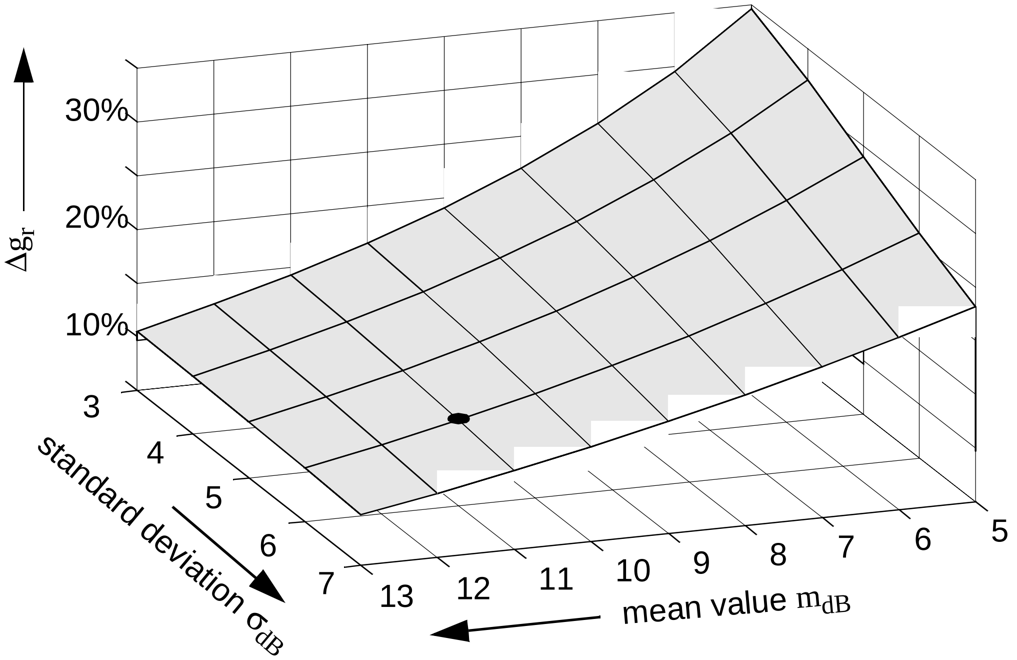


Figure 4.31. Relative throughput gain Δg_r for selected m_{dB} and σ_{dB} on a TU3-channel with ideal FH for a selective-repeat ARQ protocol; noise limited scenario.

The above simulations can also be carried out for the interference limited case, i.e., the FER are plotted versus C/I. The maximum throughput versus C/I according to ideal link adaptation for the 0th and 4th iterations, is depicted in Fig. 4.32.

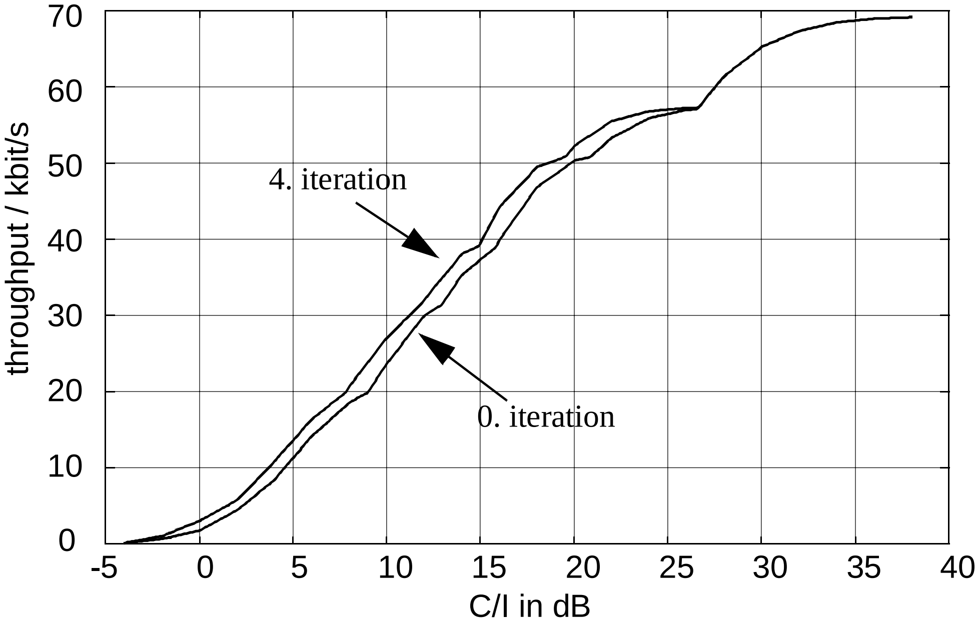


Figure 4.32. Maximum throughput versus C/I per modulated bit for PSC-1 to PCS-6 on a TU3-channel with ideal FH for a selective-repeat ARQ protocol.

Again, assuming a normal distribution for the C/I-values the relative throughput gains as depicted in Fig. 4.33 are obtained. For the interference limited case the relative throughput gain ranges from about 5% to 11%, depending on the environment.

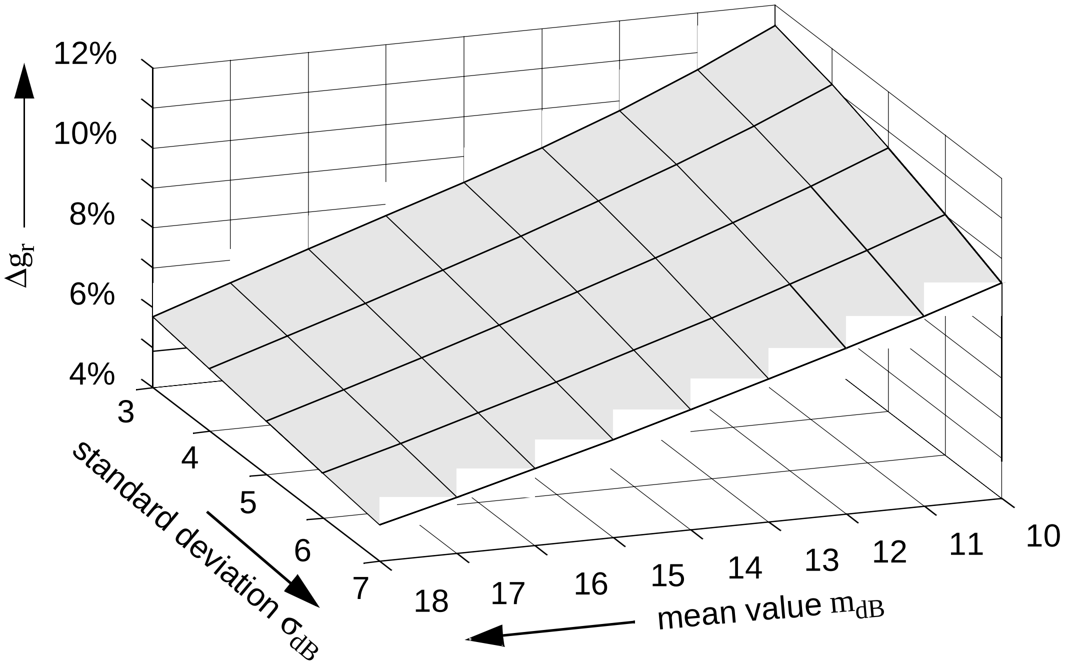


Figure 4.33. Relative throughput gain Δg_r for selected m_{dB} and σ_{dB} on a TU3-channel with ideal FH for a selective-repeat ARQ protocol; interference limited scenario.

Example 4.2 *From system level simulations the C/I-distribution can be obtained. In Fig. 4.34 the pdf of the local mean C/I values is given. The simulation parameters are a path loss of 35 dB, a system load of 75%, a re-use pattern of 1/3 and hexagonal cells. With this distribution of C/I the absolute throughput gain of 2.6 kbit/s and a relative throughput gain* Δg_r of *9% are achieved.*

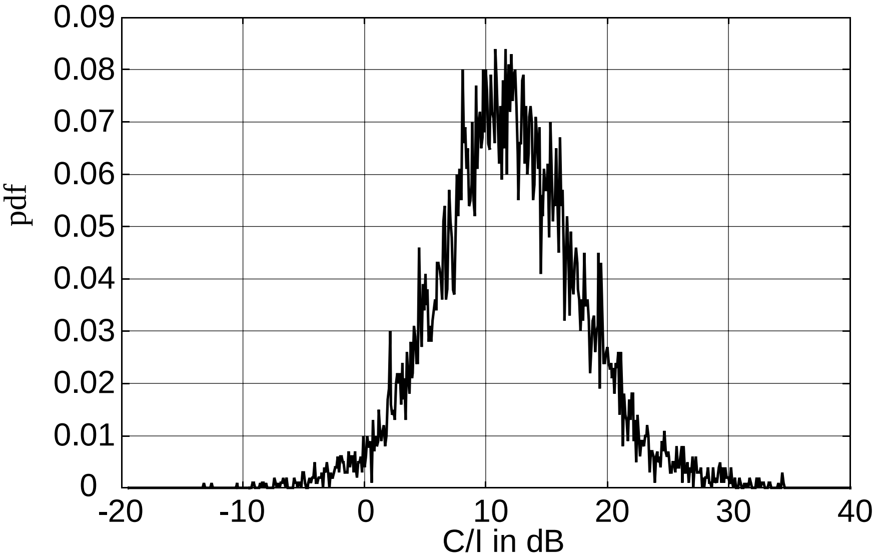


Figure 4.34. Pdf of local mean of C/I; system load: 75%, path loss slope: 35 dB per decade, re-use: 1/3, hexagonal cells.

4.2.3 Turbo-detection for enhanced circuit switched data

Since the beginning of the standardization process several coding schemes have been under discussion for ECSD [ETS98f]. As low bit error rates of 10^{-3} to 10^{-5} are to be supported concatenated coding schemes are proposed. One of the proposed coding schemes is a SCCC scheme of rate 1/2. At the time of this work the parameters of the coding and interleaving scheme were not fixed. Since the parameters are constantly changing, for purpose of illustration the coding parameters are selected which show the potential of turbo-detection for a SCCC scheme in EDGE.

In Fig. 4.35 the block diagram of the considered SCCC scheme is depicted. First, the input sequence u(D) is encoded with a non-recursive convolutional code of rate R=1/2 and generator matrix $G(D) = (1 + D^3 + D^4, 1 + D + D^3 + D^4)$. After puncturing and multiplexing, the sequence $c_i(D)$ is interleaved and encoded with a recursive convolutional code of rate R=1 and generator matrix $G(D) = (1/(1 + D))$. The indices i and o represent the inner and outer code, respectively. Note that the inner convolutional encoder is a differential encoder as used for DPSK (d̲ifferential p̲hase s̲hift-k̲eying).

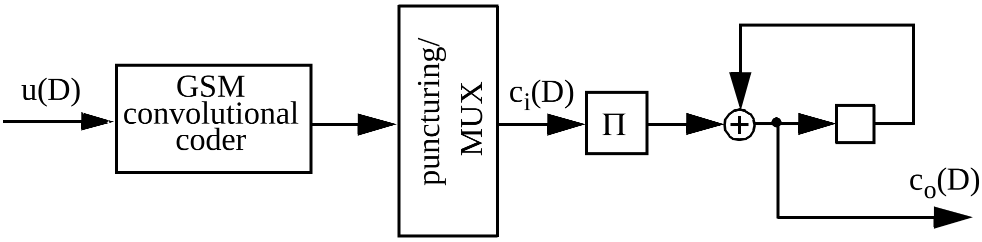


Figure 4.35. Serial concatenated convolutional encoder.

The turbo-decoder for this scheme is given in Fig. 4.36. This iterative decoding scheme is similar to turbo-detection, except that the equalizer is exchanged with the inner decoder. From the equalizer the L-values $L(\mathbf{c}_i)$ of the code bits of the inner code are obtained. The inner SISO decoder generates the soft-output $L^i_*(\hat{\mathbf{c}}_o')$ that is the extrinsic information plus the channel information[1]. After deinterleaving, the outer SISO decoder generates the a posteriori information, $L^D(\hat{\mathbf{u}})$, about the information sequence, **u,** and the extrinsic information, $L^o_e(\hat{\mathbf{c}}_o)$, denoting the turbo component. It is interleaved and used as a priori information for the next iteration. Now, the inner SISO decoder uses both the information from the equalizer, $L(\mathbf{c}_i)$, and the information fed back from the decoder, $L^o_e(\hat{\mathbf{c}}_o)$.

[1] For clarity the values $L^i_*(\hat{c}_o')$ and $L^o_e(\hat{c}_o)$ are directly calculated within the SISO decoders. The subtraction of the intrinsic values is now a part of the decoder.

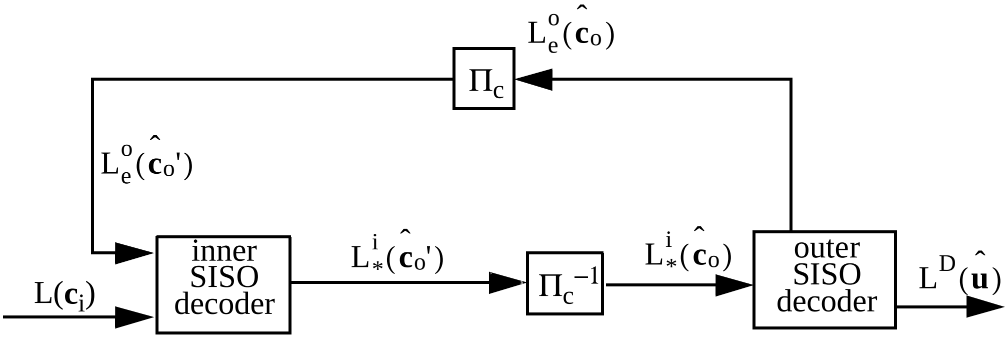
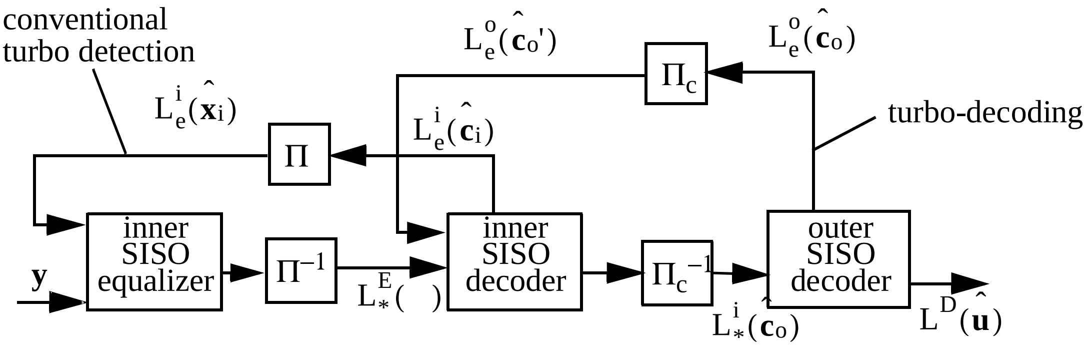

$L_e^o(\hat{\mathbf{c}}_o)$
Π_c
$L_e^o(\hat{\mathbf{c}}_o')$
$L(\mathbf{c}_i)$
inner SISO decoder
$L_*^i(\hat{\mathbf{c}}_o')$
Π_c^{-1}
$L_*^i(\hat{\mathbf{c}}_o)$
outer SISO decoder
$L^D(\hat{\mathbf{u}})$

Figure 4.36. Block diagram of iterative decoding of SCCC.

Iterative decoding can also be combined with turbo-detection, see Fig. 4.37. The mobile radio channel is again regarded as a time-varying convolutional encoder with complex symbols. Hence, the SCCC encoder together with the mobile radio channel can be considered as a double-serially concatenated code system which can be decoded iteratively [BDM98a]. Note that turbo-detection can also be performed with PCCC (parallel concatenated convolutional code) schemes [RaZ97].

The channel values $\mathbf{y}$ are equalized and deinterleaved. The sequence $L_*^E(\hat{\mathbf{c}}_i')$ is decoded by the inner decoder which generates two outputs. The first generated is the extrinsic values $L_e^i(\hat{\mathbf{c}}_i)$ of the coded bits of the inner code. These values are not used during this iteration. The second output is the values $L_*^i(\hat{\mathbf{c}}_o')$ of the code bits of the outer code. These values are deinterleaved and passed to the outer decoder. The outer decoder calculates the a posteriori information, $L^D(\hat{\mathbf{u}})$, about the information bits. These values can be directly passed to the data sink. The first iteration is complete. The outer decoder has also generated the extrinsic information, $L_e^o(\hat{\mathbf{c}}_o)$, for the outer code bits. These values denote the turbo-component of the turbo-decoder and are interleaved together with the values, $L_*^E(\hat{\mathbf{c}}_i)$, used by the inner SISO decoder. The inner SISO decoder generates the extrinsic values, $L_e^E(\hat{\mathbf{c}}_i)$, for the inner code bits and the L-values, $L_*^i(\hat{\mathbf{c}}_o')$, for the outer code bits. Instead of forwarding the L-values, $L_*^i(\hat{\mathbf{c}}_o')$, to the outer decoder as would be done in turbo-decoding, the information $L_e^E(\hat{\mathbf{c}}_i)$ denoting the second turbo-component is fed back to the equalizer. After interleaving, $L_e^i(\hat{x}_i)$ is used as a priori information for the equalization in the next iteration. Having equalized the bursts again, the deinterleaved values $L_*^E(\hat{\mathbf{c}}_i)$ and $L_e^o(\hat{\mathbf{c}}_o')$ are used for the inner decoder. This time the values $L^i_e(\hat{\ })$ are not used; instead the values $L^i_*(\hat{\ })$ are processed. This iterative procedure can be repeated to improve the error performance.



Π^{-1}: channel deinterleaver Π_c^{-1}: deinterleaver for code
Π: channel interleaver Π_c: interleaver for code

Figure 4.37. Block diagram of turbo-detection including turbo-decoding of SCCC.

In Fig. 4.39 the BER after the decoder versus E_b/N_0 for several iterations is depicted. The parameters of the simulation are chosen as follows:

- data block length: 688 bits, 4 tail bits are appended;
- outer encoder: rate R=1/2 convolutional coder with generator matrix $G_o(D) = (1 + D^3 + D^4, 1 + D + D^3 + D^4)$;
- coded frame length: 1384 bits;
- code interleaver: 8 coded frames, comprising 11072 bits, are row-column interleaved; the row-column interleaver designed to generate a maximum Block Hamming distance[1];
- inner encoder: rate R=1 recursive convolutional encoder with the generator polynomial $G_i(D) = (1/(1 + D))$;
- channel interleaving: 1384 bits are block rectangular interleaved and formatted in four bursts, similar to EGPRS;
- the transmission channel: the channel impulse response is given by $h(t) = g(t)a(t) = (0.5j\delta(t+T) + \delta(t) - 0.5j\delta(t-T))a(t)$, g(t) is time-unvarying and a(t) is time-varying, modeled by a one-tap correlated Rayleigh fading corresponding to 3 km/h on 900 MHz, ideal FH is assumed.

[1] If the two interleavers are not matched the performance of the system degrades.

In Fig. 4.38 the BER is depicted applying turbo-detection combined with turbo-decoding.

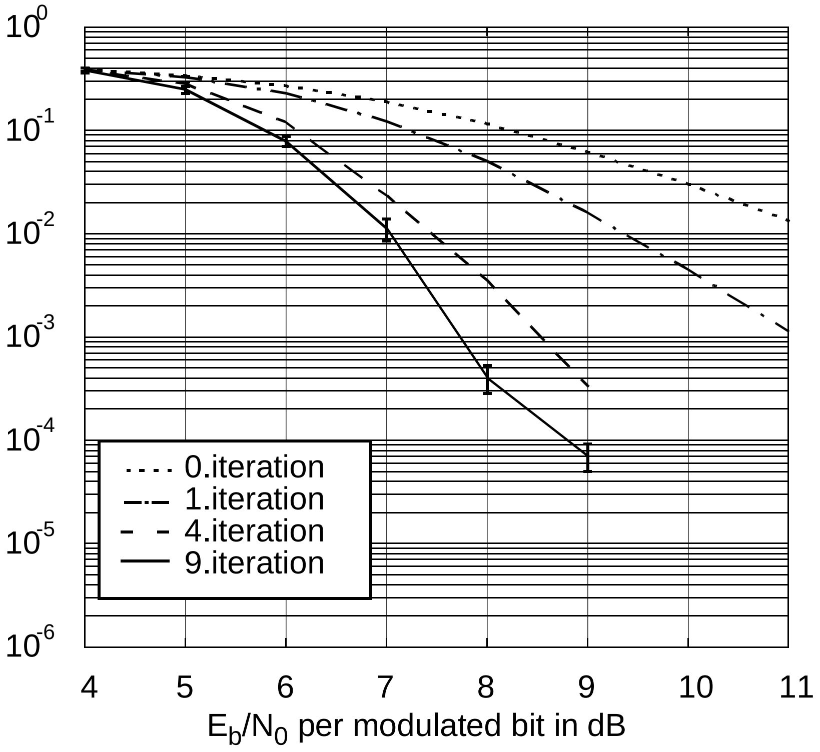


Figure 4.38. BER after the decoder versus E_b/N_0 per modulated bit for PCS-1 on a one-tap correlated fading channel with ideal FH, correlated fading corresponding to 3 km/h on 900 MHz; turbo-detection including turbo-decoding is applied.

The solid lines in Fig. 4.39 give the BER if only turbo-decoding is applied. The dashed lines give the BER for turbo-detection as illustrated in Fig. 4.37. It can be seen that the latter scheme outperforms the conventional turbo-decoding scheme by 1.3 dB and 2 dB after the 9th iteration, respectively, at a BER of 10^{-3} and 10^{-4}. For both schemes an error floor is observed. For PCCC codes this error floor is typical [Rob94]. It should-should be lower for serially concatenated convolutional codes. It can be lowered by using different interleavers, larger blocks and different constituent encoders [BeM98b] or woven codes [JBS98]. As it is not the purpose of this work to optimize the code parameters for turbo-codes, it suffices to show that by turbo-detection the performance of the turbo-coding scheme can be further improved.

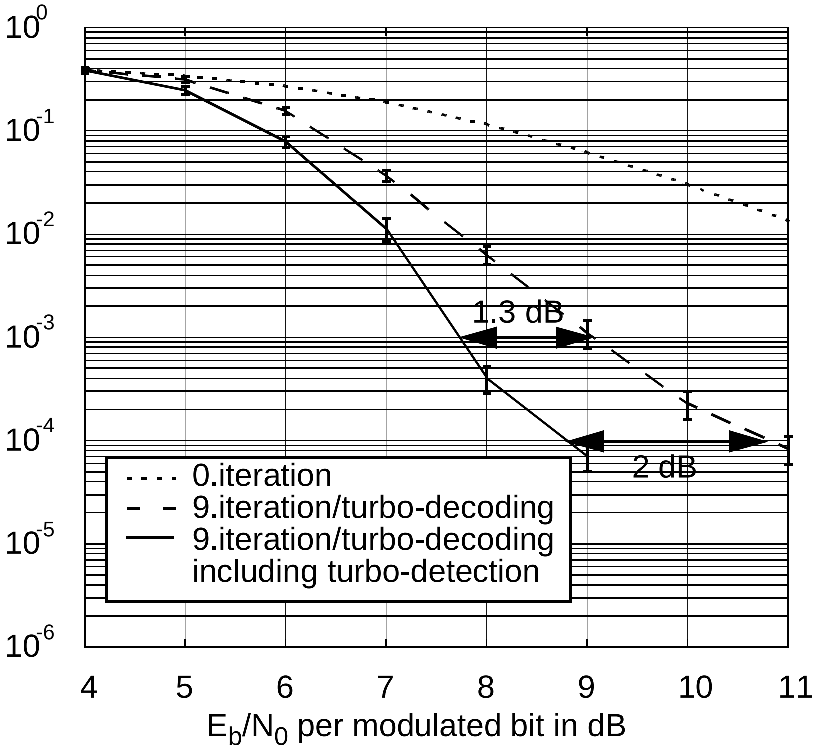


Figure 4.39. BER after the decoder versus E_b/N_0 per modulated bit for PCS-1 on a one-tap correlated fading channel with ideal FH, correlated fading corresponding to 3 km/h on 900 MHz.

5 Adaptive channel re-estimation

In Chapter 4 it was shown that for GSM systems using GMSK modulation large iteration gains can be achieved only for the BER after the equalizer. Due to the orthogonal ISI of GMSK modulation, these improvements cannot be realized after the decoder; for the decoded bits no significant iteration gains in the BER and FER performance can be achieved. Only for the 8-PSK modulated services of EDGE significant iteration gains can be achieved.

In previous chapters non-adaptive channel estimation techniques have been considered. The channel parameters have been estimated from the midamble and are fixed for the remainder of the burst. On fast fading channels, i.e. when there are significant time variations of the channel within a half burst, the channel estimates become inaccurate, particularly at the edges of the burst. The error performance of the detector suffers due to the mismatched channel equalization. These effects and their impacts on system performance are discussed in Section 5.1.

Adaptive channel estimation techniques can be applied to track the time variance and to counteract the performance decrease [Qur85]. For adaptive channel estimation, as is explained in Section 5.2, the equalizer decisions are used to update channel estimates adaptively for each received value of the entire burst [MaP73, Ung74]. If the equalizer decisions are incorrect, adaptive channel estimation suffers. The tracking algorithm is not able to follow the time variation of the channel.

Turbo-detection improves the equalizer decisions even for GMSK modulated signals where no significant iteration gains are observed after the decoder. In Section 5.3 a novel detection scheme is developed which exploits this effect. By incorporating adaptive channel estimation into the turbo-loop, the improved equalizer decisions can then be used to enhance the channel tracking capabilities and, hence, improve the equalizer performance. The more accurate soft-output values of the equalizer improve the BER after the decoder as well. As the channel parameters are reestimated for each iteration anew, this technique is called CRE (channel re-estimation).

5.1 The effects of non-adaptive channel estimation on time-variant channels

Channel parameter estimation and equalization was introduced in Section 2.4. If the channel changes slowly, it can be assumed that the variations of the channel parameters are negligible. In this situation it is appropriate to estimate the channel parameters from the midamble and to keep the estimated channel parameters constant for the remainder of the burst. However, as the channel changes faster, the channel estimates become inaccurate, particularly at the edges of the bursts, and the performance of the equalizer suffers. In Fig. 5.1 the distribution of bit errors over normal bursts is shown for a

RA250 channel. The channel parameters are estimated from the midamble and are fixed for the entire burst. Towards the ends of the burst the number of bit errors after the equalizer increases, resulting in a degradation of the performance.

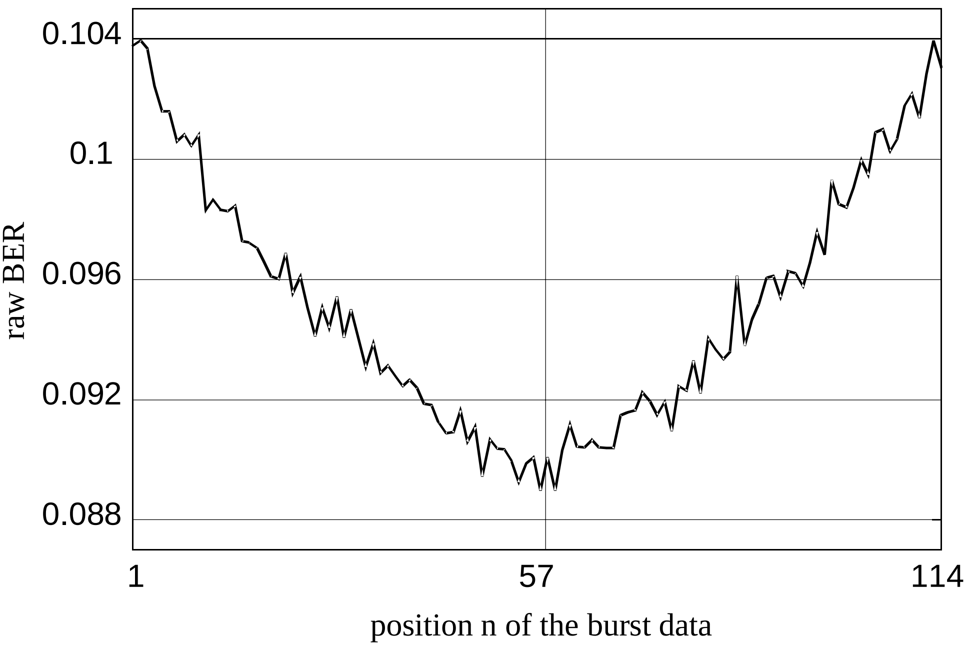


Figure 5.1. Distribution of bit errors over the burst for a RA250 channel without adaptive channel estimation at E_b/N_0= 4 dB.

In order to illustrate the inaccuracy of the channel estimates, the mean tap weight error is introduced and is denoted by

$$t_n = \overline{\left\|\mathbf{h}_n - \hat{\mathbf{h}}_n\right\|^2}, \tag{5.1}$$

with $\mathbf{h}_n$ being the channel vector for the n^{th} bit of the burst, and $\mathbf{h}_n$ the corresponding estimated channel vector. Note that for non-adaptive channel estimation, $\mathbf{h}_n$ is fixed for all values of n; the mean value over all transmitted bursts is assumed. Normalizing the mean tap error t_n by the mean tap error t_m in the middle of the midamble, the normalized mean tap error $t_{n,m} = t_n / t_m$ is obtained.

Fig. 5.2 gives the normalized mean tap weight error $t_{n,m}$ versus the burst position for a single-tap fading channel with a maximal Doppler frequency of $f_{D,max} = 208.33\text{Hz}$ corresponding to a velocity v of 250 km/h at a carrier frequency f_c of 900 MHz. The single-tap fading channel rotates about 15 degrees during one half-burst. At the outer symbols of the burst the estimated channel coefficients diverge strongly from the channel coefficients.

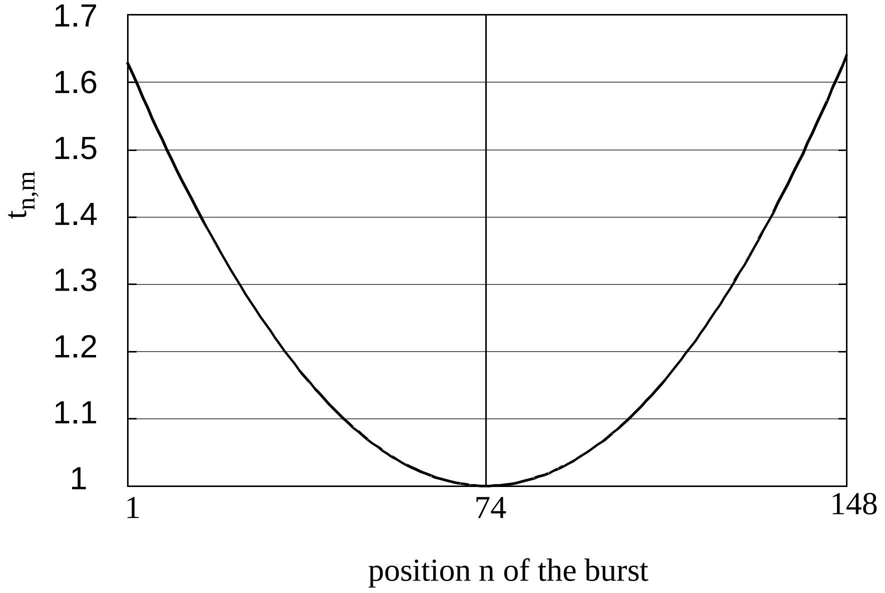


Figure 5.2. Normalized mean tap weight error $t_{n,m}$ versus the burst position for a one-tap fading channel without adaptive channel estimation, the TX and RX filter are modelled as $\mathbf{h}_n = \begin{bmatrix} 0.5j & 1 & -0.5j \end{bmatrix}^T$.

In order to give an idea on the performance decrease of mismatched channel equalization in GSM, TCH/FS transmission via a RA250 channel without FH is simulated in two different ways. In the first scenario, the channel taps are changed during the burst, and again from burst to burst. In the second scenario, the channel taps are only changed from burst to burst and are constant during the burst. In the latter case the effects of mismatched channel equalization, due to time-variance, are excluded. The comparison of the performance in both scenarios gives the magnitude of the loss caused from mismatched channel equalization. In Fig. 5.3 the FER after the decoder for both scenarios in a RA250 environment without FH are given. For equalization and decoding the MAX-Log-MAP algorithm is used[1]. The non-tracked time-variations of the channel cause a performance decrease of approximately 0.9 dB at a FER of 3%. In Fig. 5.4 the same results are shown for a RA500 environment without FH. With increasing velocity, the degradation due to non-adaptive channel estimation grows. At a FER of 3% the performance decrease amounts to approximately 1.8 dB.

[1] On fast time-variant channels, channel parameter estimation is inaccurate. Since the Log-MAP algorithm is more sensitive to erroneous channel parameter estimates, the Max-Log-MAP equalizer outperforms the MAP equalizer on these types of channels.

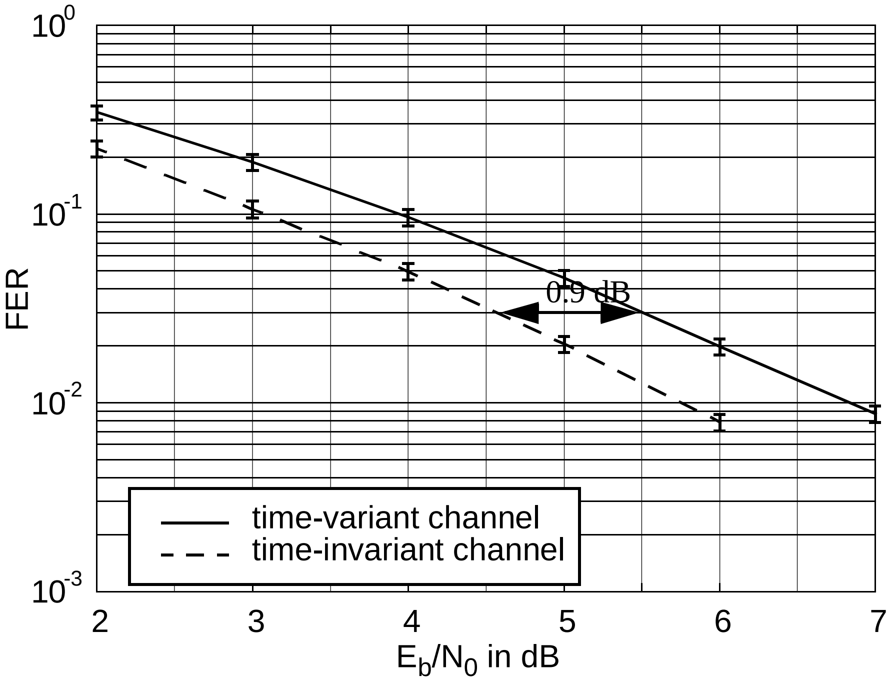


Figure 5.3. FER for RA250 (no FH) with non-adaptive CE.

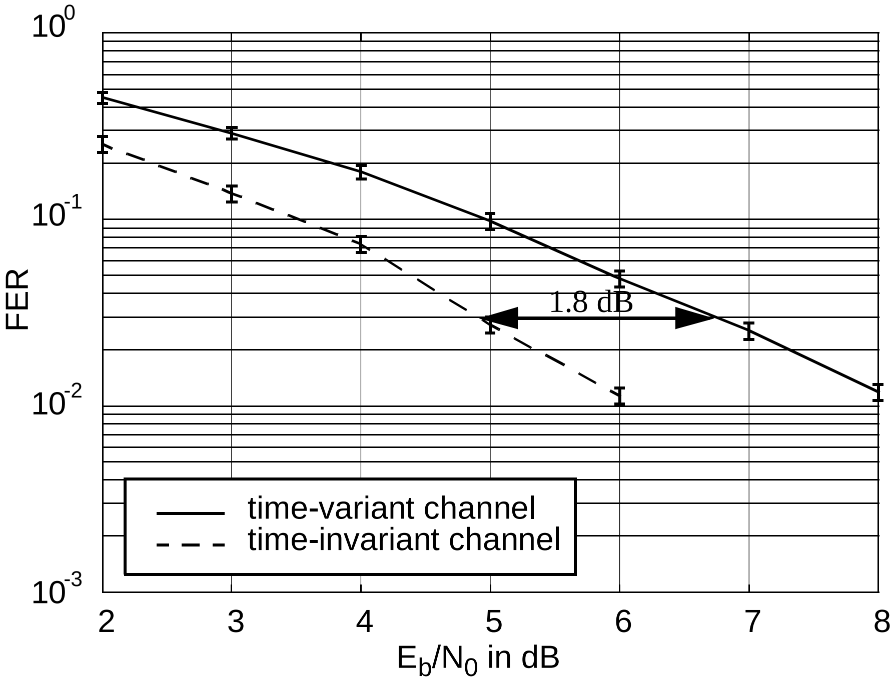


Figure 5.4. FER for RA500 (no FH) with non-adaptive CE.

In the following a method for counteracting this degradation is presented.

5.2 The general principles of adaptive channel estimation and equalization

In Section 5.1 the effects of non-adaptive CE and equalization on time-varying channels were given. In order to compensate for the negative effects of fast time-variations, adaptive CE and adaptive equalization must be employed. Before considering the incorporation of adaptive CE and adaptive equalization into turbo-detection, two main principles are explained: adaptive CE and adaptive MLSE.

5.2.1 Adaptive channel estimation

There are two main algorithms used for adaptive CE in TDMA systems [Pro91], the RLS (recursive least squares) algorithm and the LMS (least mean square) algorithm. Although the RLS algorithm converges faster, the LMS algorithm has better tracking capabilities [Hay96]. The LMS algorithm is more robust and has a lower complexity. Since in GSM-systems an initial channel estimate is obtained from the midamble, only channel tracking is required. Thus, the LMS algorithm is better suited for GSM-systems [NeM91] and, hence, the LMS algorithm is considered in the following.

The delay spread of most GSM-environments is short, e.g. most of the energy is received within one symbol duration T. Since the system is not able to dissolve paths arriving within a delay smaller than T, the fading environment is typically represented by one single fading tap in the receiver. This particular channel characteristic can be taken into account by introducing additional constraints in the LMS algorithm. In this thesis a method is presented that employs an additional common factor for adaptive algorithms [Rad96]. This rotator LMS is given in Appendix B. This particular algorithm is incorporated in the following adaptive channel estimation scheme.

In Fig. 5.5 the principle of adaptive channel estimation is depicted. The following steps are processed for each received value y_n:

- First, the detected symbols, $x_n,...,x_{n-L+1}$, are convolved with a transversal filter that rebuilds the estimated channel impulse response $\mathbf{h}_n$. The resulting reference value y_n is subtracted from the received values. This difference constitutes the error e_n.
- Second, the resulting error e_n, the current channel estimate $\mathbf{h}_n$ and the detected symbols, $x_n,...,x_{n-L+1}$, are used to update the channel estimate according to an adaptation algorithm, e.g. the rotator LMS algorithm. Then, the filter coefficients of the transversal filter are replaced with the new channel estimate $\mathbf{h}_{n+1}$.

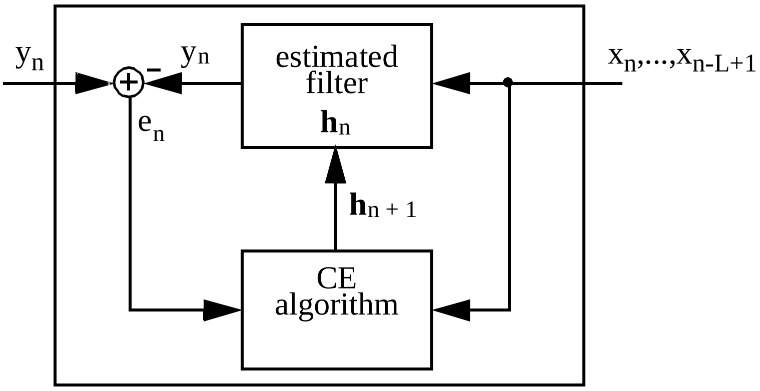


Figure 5.5. Adaptive channel estimator.

5.2.2 Adaptive maximum likelihood sequence estimation

In order to estimate the channel impulse response adaptively, the transmitted symbols $x_n,...,x_{n-L+1}$ have to be passed to the channel estimator. At the receiver only the transmitted bits of the midamble are known. Since the transmitted information symbols $x_n,...,x_{n-L+1}$ are not known at the receiver, the tentative decisions, $x_n, \ldots, x_{n-L+1}$, of the equalizer have to be passed to the adaptive channel estimator.

In [MaP73], the scheme of adaptive MLSE is proposed. MLSE is used in conjunction with adaptive CE. The adaptive MLSE is illustrated in Fig. 5.6. After each receive value y_n, the MLSE algorithm, i.e. the VA, provides an estimate x_{n-d} of the transmitted symbol with a delay d. Note that d is the decision depth of the VA. The previous L decisions are fed to the channel estimator where the error $e_{n\text{-}d}$ is generated and the new channel impulse response $\mathbf{h}_{n-d}$ is calculated. This new channel impulse response $\mathbf{h}_{n-d}$ is passed to the equalizer, and the metric increments for the next receive value are calculated using this updated channel impulse response. Consequently, the equalizer adapts to the new channel conditions.

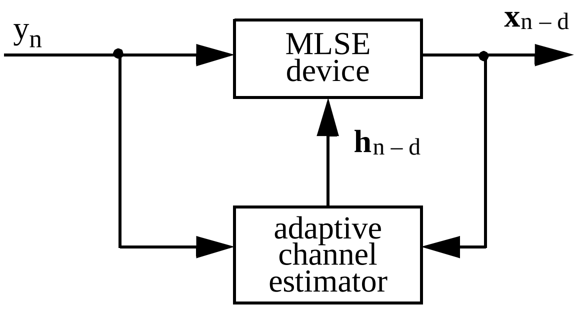


Figure 5.6. Adaptive MLSE scheme.

On one hand, from [HeJ71] it is known that the performance of the VA suffers if the decision depth d is not large enough. In order to get a close to optimum performance, the decision depth d should be five times the constraint length κ of the code. On the other hand, in adaptive MLSE the channel estimates can only be updated with a delay of d. If the delay is too large, the adaptive device cannot track the channel variations accurately. Therefore, a compromise must be found. This can be done by simulations. In this thesis a decision depth of 15 is used. This decision depth is only three times the constraint length of the channel, degrading the tentative decisions. However, as will be shown in the next section, the tentative decisions are not passed to the decoder. Hence, its impact on the system performance is not large.

5.3 Adaptive equalization for turbo-detection

For turbo-detection a SISO equalizer has to be used. The scheme introduced in Section 5.2 can be easily modified to generate soft-output by using the SOVA instead of the VA. However, the best iteration gains of turbo-detection can be achieved using the MAP or the Max-Log-MAP algorithm [BaF97b]. Due to the strong sensitivity of the MAP algorithm to non-ideal channel knowledge, the Max-Log-MAP algorithm is better suited for equalization on fast time-varying channels. Therefore, in this chapter adaptive MLSE is replace with adaptive Max-Log-MAP equalization in order to obtain an appropriate adaptive SISO equalizer.

For adaptive Max-Log-MAP equalization, the Max-Log-MAP algorithm is modified. In order to update the channel estimates adaptively, the Max-Log-MAP algorithm must generate tentative decisions. Therefore, the Max-Log-MAP algorithm must store the path history of the survivor path. The tentative decision $\mathbf{x}_{n-d}$ is the bit n-d of the path corresponding to the most probable state S_n. With this modification the forward recursion of the Max-Log-MAP algorithm is identical to the MLSE algorithm.

This modified Max-Log-MAP equalizer can now be combined with adaptive CE. The following changes have to be applied to the adaptive MLSE scheme in order to obtain the adaptive SISO equalizer, see Fig. 5.6 and Fig. 5.7:

- Instead of the adaptive MLSE device, the adaptive Max-Log-MAP equalizer is employed.
- The tentative decisions $\mathbf{x}_{n-d}$ are used for adaptive channel estimation only.
- Instead of hard-values, the soft-values $L_*^E(\hat{x})$ are generated.
- A priori information L($\mathbf{x}$) about the symbols to be equalized are obtained from the decoder. This information is utilized by the adaptive Max-Log-MAP equalizer.

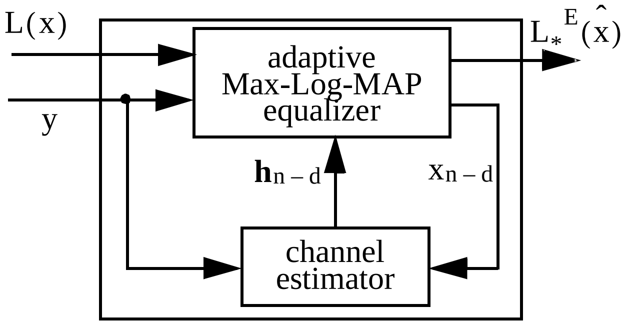


Figure 5.7. Adaptive SISO equalizer.

For turbo-detection the adaptive SISO equalizer of Fig. 5.7 is utilized as depicted in Fig. 5.8. Similar to the method described in Section 4.2.3, the values $L_*^E(\hat{\mathbf{x}})$ and $L_e^D(\hat{\mathbf{c}})$ are directly generated inside the SISO modules.

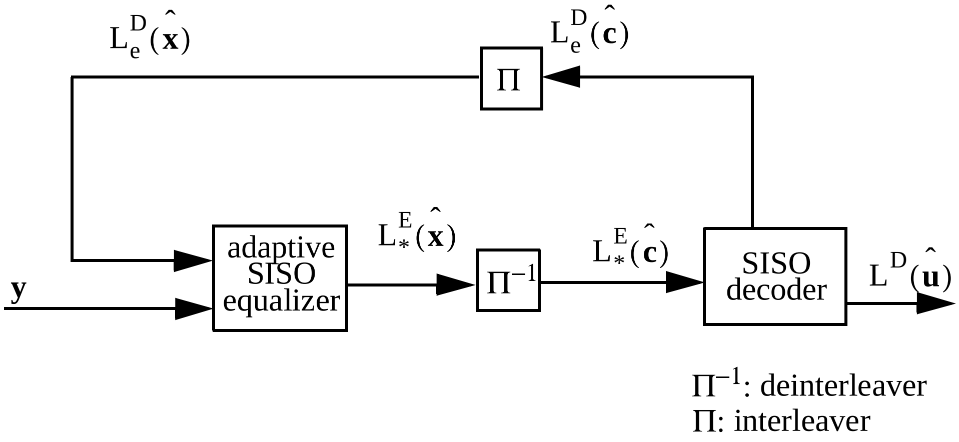

$L_e^D(\hat{\mathbf{x}})$
$L_e^D(\hat{\mathbf{c}})$
Π
adaptive
SISO
equalizer
$L_*^E(\hat{\mathbf{x}})$
Π^{-1}
$L_*^E(\hat{\mathbf{c}})$
SISO
decoder
$L^D(\hat{\mathbf{u}})$
y
Π^{-1}: deinterleaver
Π: interleaver

Figure 5.8. Block diagram of iterative channel estimation, equalization and decoding.

Before beginning the investigation of the performance of the adaptive SISO equalizer in combination with turbo-detection, the performance of the adaptive SISO equalizer is analysed with non-iterative detection. Therefore, two environments, as described in Section 5.1, are examined. In the first scenario the channel impulse responses are kept fixed throughout the whole burst and are only changed from burst to burst. In the following this environment is referred to as time-invariant from the point of view of the equalizer. The performance without degradation due to speed can be investigated. In the second scenario the channel impulse responses are continuously changed for each symbol. In Fig. 5.9 the raw BER for both scenarios are depicted for a RA250 (no FH) channel. Non-adaptive equalization is compared to adaptive equalization. It can be seen that with the adaptive equalizer about the same raw BER as for the time-invariant channel is obtained. The performance of the adaptive scheme is about 0.7 dB better at a raw BER of 8% than the non-adaptive scheme. In Fig. 5.10 and Fig. 5.11 the corresponding BER of the decoded class 1 bits and the FER are illustrated, respectively.

Both the FER and the BER (class 1) come close to the performance of the time-invariant case. Compared to the non-adaptive detection scheme the BER (class 1) and the FER are approximately 0.8 dB better at a BER and a FER of 3%. Although the same performance of the time-invariant channel cannot be achieved, the adaptive equalizer nearly compensates for the degradation due to motion. The performance differences are within the simulation inaccuracies.

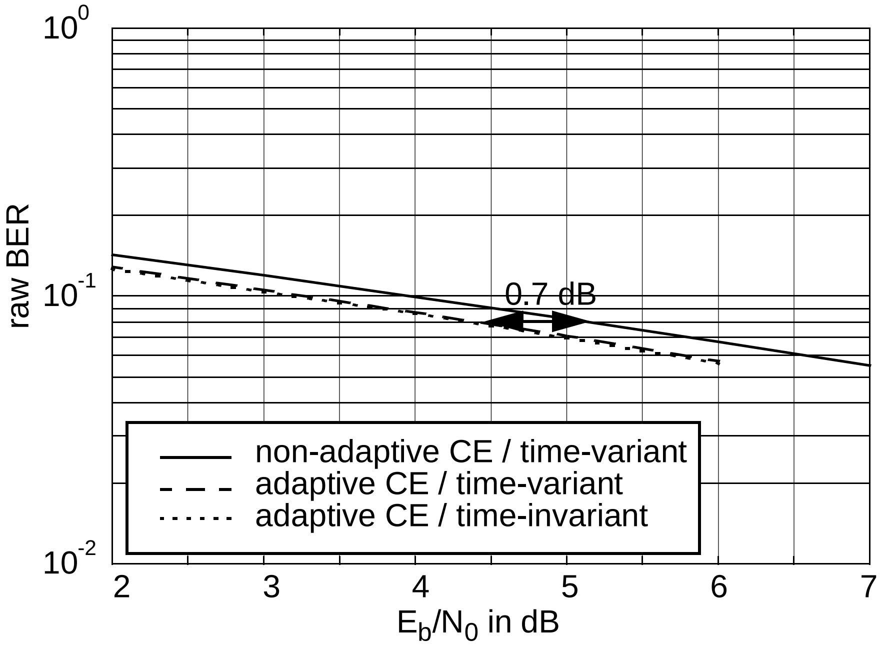


Figure 5.9. Raw BER for RA250 (no FH) with non-adaptive CE and adaptive CE for time-variant and time-invariant channel

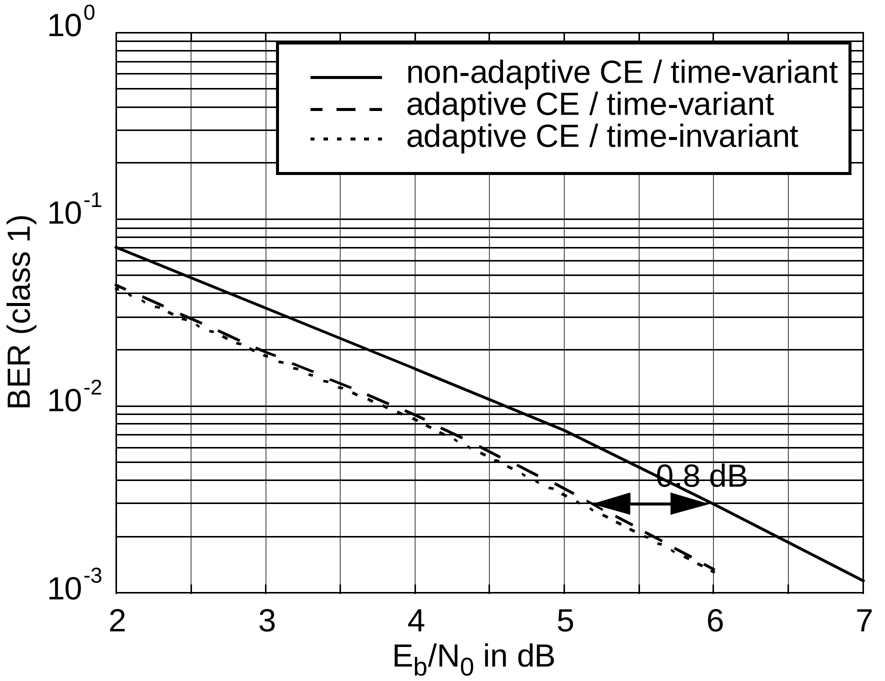


Figure 5.10. BER (class 1) after the decoder with non-adaptive CE and adaptive CE for time-varying and time-invariant RA250 (no FH) channel.

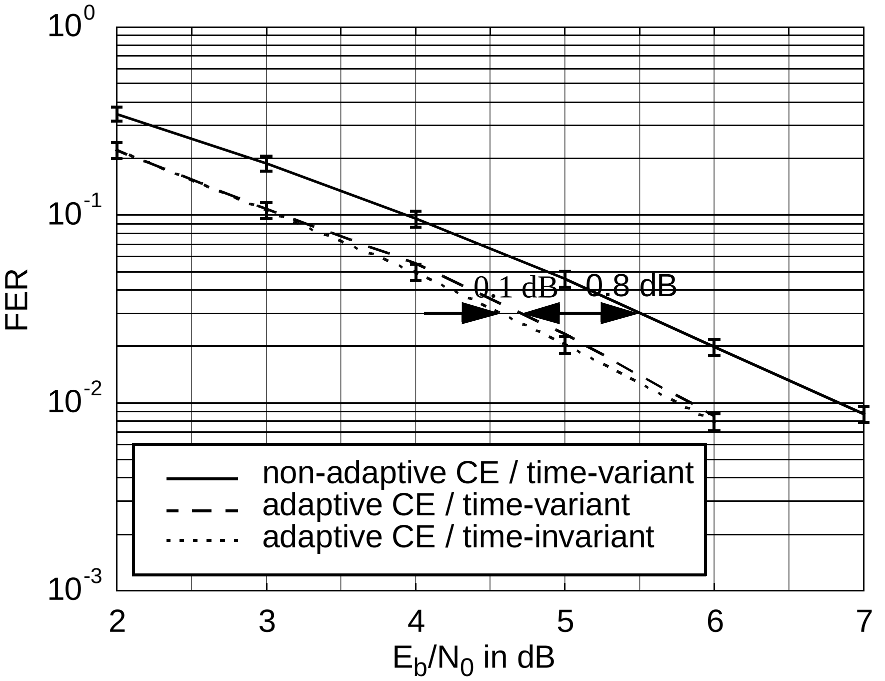


Figure 5.11. FER with non-adaptive CE and adaptive CE for time-variant and time-invariant RA250 (no FH) channel.

In turbo-detection the a priori information $L_e^D(\hat{\mathbf{x}})$ from the decoder affects the calculation of the logarithm of the transition probabilities $\Gamma_{i,0}(s', s)$ in the equalizer. These transition probabilities were given in Section 2.4 as:

$$\Gamma_{i,0}(s', s) = \ln\{\gamma_{i,0}(s', s)\} = \frac{1}{2}L_e^D(\hat{x}_i) - \frac{|y_i - c_{s',0}|^2}{2\hat{\sigma}^2}. \tag{5.2}$$

From (5.2) it is obvious that the transition probabilities are affected in two ways:

- The tentative decisions $\mathbf{x}_{n-d}$ become more reliable and, hence, the channel estimates are more accurate. This improves the calculation of the reference channel symbols $c_{s',0}$ and affects the second term of (5.2).
- The values $L_e^D(\hat{\mathbf{x}})$, denoting the first term of (5.2), directly influence the transition probabilities $\Gamma_{i,0}(s', s)$.

In order to investigate both effects separately, the scheme depicted in Fig. 5.12 can be used.

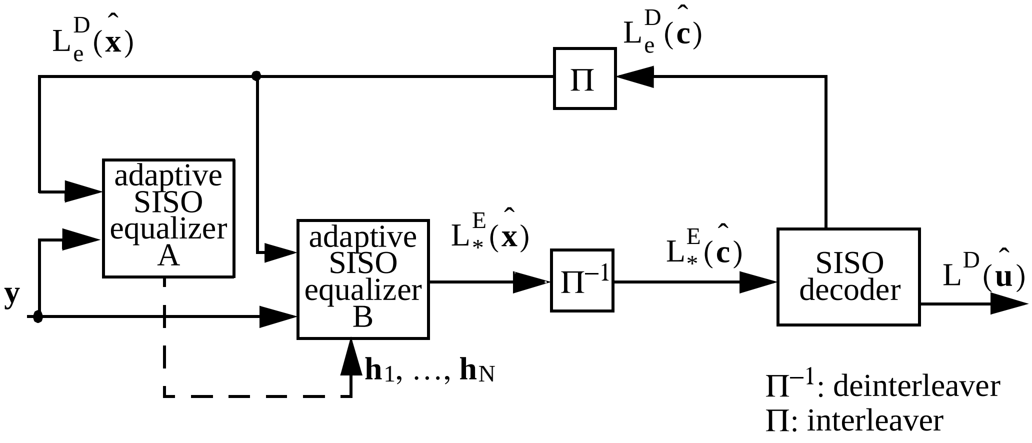

$L_e^D(\hat{\mathbf{x}})$
$L_e^D(\hat{\mathbf{c}})$
Π
adaptive SISO equalizer A
adaptive SISO equalizer B
$L_*^E(\hat{\mathbf{x}})$
Π^{-1}
$L_*^E(\hat{\mathbf{c}})$
SISO decoder
$L^D(\hat{\mathbf{u}})$
$\mathbf{y}$
$\mathbf{h}_1, \ldots, \mathbf{h}_N$
Π^{-1}: deinterleaver
Π: interleaver

Figure 5.12. Block diagram of iterative channel estimation, equalization and decoding.

The only difference from conventional turbo-detection is that the adaptive SISO equalizer is used twice. Adaptive SISO equalizer A generates the adaptively updated channel estimates $\mathbf{h}_1, \ldots, \mathbf{h}_N$. These estimated channel impulse responses are then utilized by adaptive SISO equalizer B. Adaptive SISO equalizer B is only adaptive in the sense that it recalculates the channel reference symbols for each step. The channel estimates themselves are not updated.

The following configurations are possible:

- Configuration A: If the information $L_e^D(\hat{x})$ is only fed to adaptive SISO equalizer A, the effect of improved CE can be investigated.

- Configuration B: If the information $L_e^D(\hat{\ })$ is only fed to adaptive SISO equalizer B, the conventional turbo-detection gain is observed without the impact of the improved channel estimates from channel re-estimation.
- Configuration C: If the information $L_e^D(\hat{\ })$ is used by both equalizer modules, the scheme is identical to that of Fig. 5.8.

In order to evaluate the benefits of the new scheme for GSM, the performance for the TCH/FS scheme is again examined.

First, the effect of the improved CE is examined using the turbo-component only for adaptive SISO equalizer A, i.e., configuration A. The raw BER are given in Fig. 5.13 for non-adaptive CE and for adaptive CE after the 0th and after the 1st iterations. Scheme 1b is used for turbo-detection. The mobile radio channel is a RA250 (no FH) channel. Compared to non-adaptive CE, the raw BER of adaptive CE is already improved by 0.7 dB during the 0th iteration. In the 1st iteration, the BER after the equalizer is further improved by 1.0 dB. This increased performance stems only from improved channel estimates.

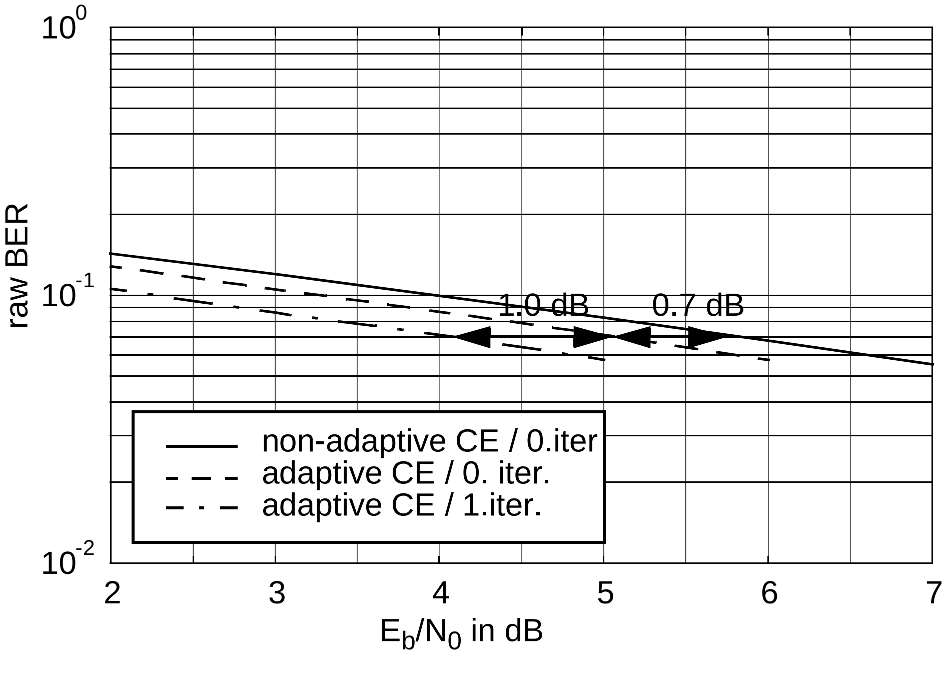


Figure 5.13. Raw BER for RA250 (no FH) for turbo-detection with non-adaptive CE and adaptive CE.

For the decoder the situation is different as is shown in Fig. 5.14 and Fig. 5.15. Even though the BER after the equalizer is improved by 1 dB, the iteration gain from the improved channel estimates shrinks to 0.2 dB for the BER (class 1) and for the FER after the decoder. The 1 dB gain of the raw BER is not maintained for the BER (class 1).

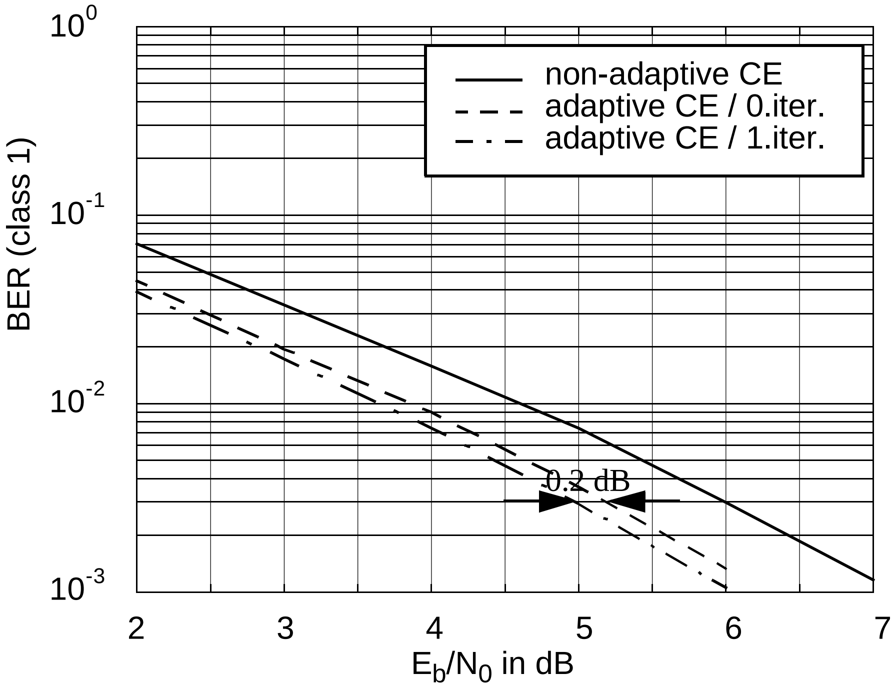


Figure 5.14. BER (class 1) for RA250 (no FH) with non-adaptive CE and with adaptive CE.

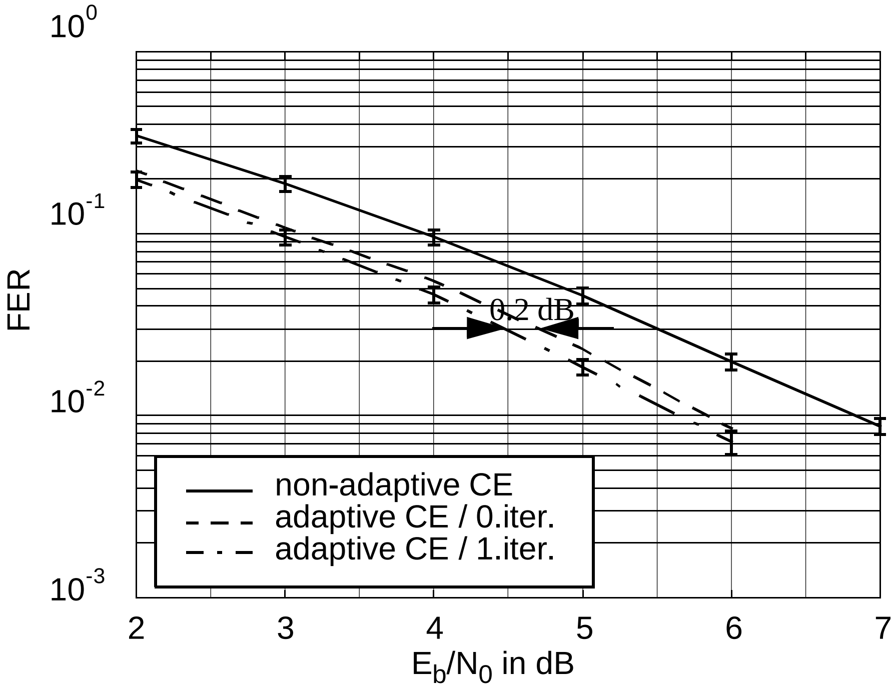


Figure 5.15. FER for RA250 (no FH) with non-adaptive CE and with adaptive CE.

The reason for this is that the channel estimates are mainly improved for bursts for which the decoder could already successfully decode the frame. Correct information fed back can improve the equalizer decisions; however, since the decoder decisions were already correct, they cannot be improved for these frames. The same behaviour is observed for the FER after the decoder, as depicted in Fig. 5.15. The gain of the 1st iteration amounts to 0.2 dB.

The improved channel estimates can be visualized by examining the normalized mean tap weight error $t_{n,m}$. Therefore, the same transmission channel as in Fig. 5.2 is used, that is a time-invariant three-tap channel multiplied with a time-varying one-tap Rayleigh fading channel with a Doppler frequency corresponding to 250 km/h at a carrier frequency of 900 Mhz. It is shown in Fig. 5.16 that during the 0th iteration the adaptive CE is first able to improve the channel estimates from the midamble. Towards the ends of the burst, tracking suffers because of erroneous preliminary decisions and the error $t_{n,m}$ increases. On the 1st iteration the error $t_{n,m}$ is decreased up to the ends of the burst. The tracking capability of the new scheme is improved.

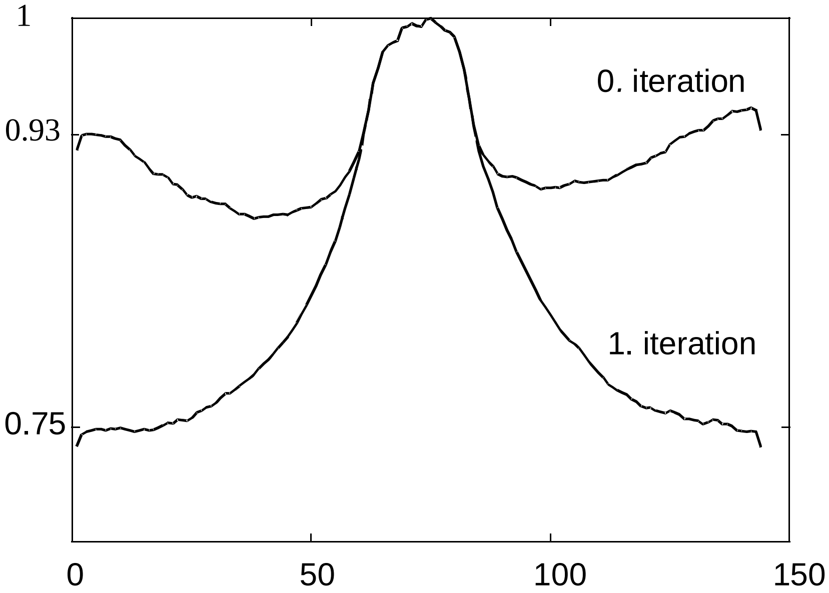


Figure 5.16. Normalized mean tap weight error $t_{n,m}$ versus the burst position for a one-tap fading channel without adaptive channel estimation. The TX and RX filters are modelled as $h_n[n] = 0.5j \cdot \delta[n] + \delta[n-1] - 0.5j \cdot \delta[n-2]$.

The iteration gains due to improved channel estimation were examined in the above simulations. In the following the turbo-component is only fed back to adaptive SISO equalizer B, i.e. configuration B. In Fig. 5.17 the FER are depicted after the 0th and the 1st iterations for scheme 2. The FER after the 1st iteration for scheme 1b is not given since it is closer to the performance after the 0th iteration. The simulations are performed for a RA250 (no FH) channel. There is nearly no iteration gain if the turbo-component is not passed to the channel estimator.

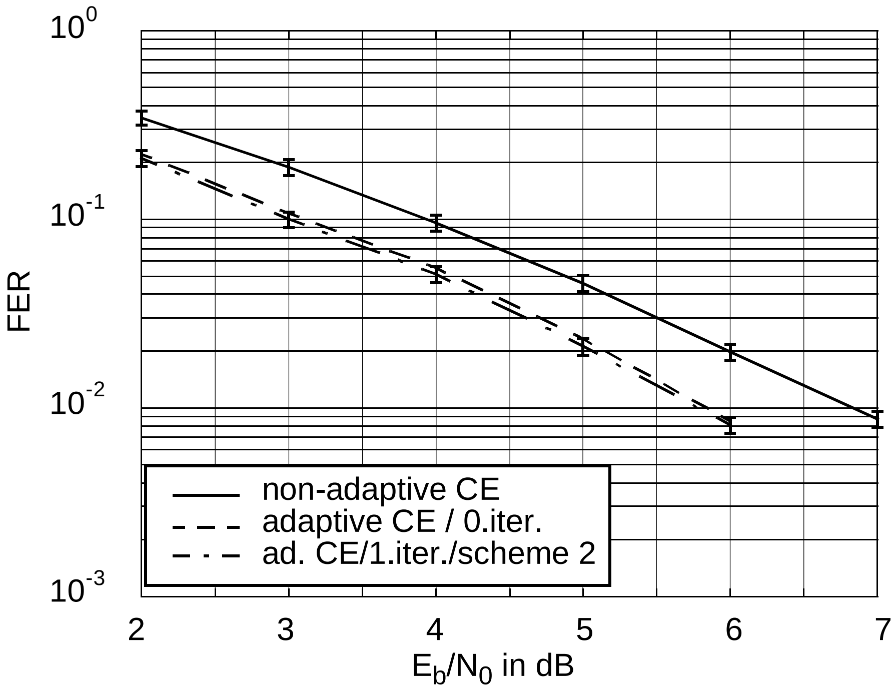


Figure 5.17. FER for RA250 (no FH) with non-adaptive CE and with adaptive CE.

The two impacts of the turbo-component were treated separately. In the following the turbo-components are fed to both adaptive SISO equalizers. Hence, the performance of turbo-detection using an adaptive SISO equalizer is obtained. Fig. 5.18 shows the FER after the 0th and the 1st iteration for scheme 1b and scheme 2. The results of further iterations are not depicted because no additional iteration gains are obtained. Scheme 2 has the best performance. The iteration gain amounts to 0.4 dB. Note that this iteration gain is only achieved because of the improved channel estimates.

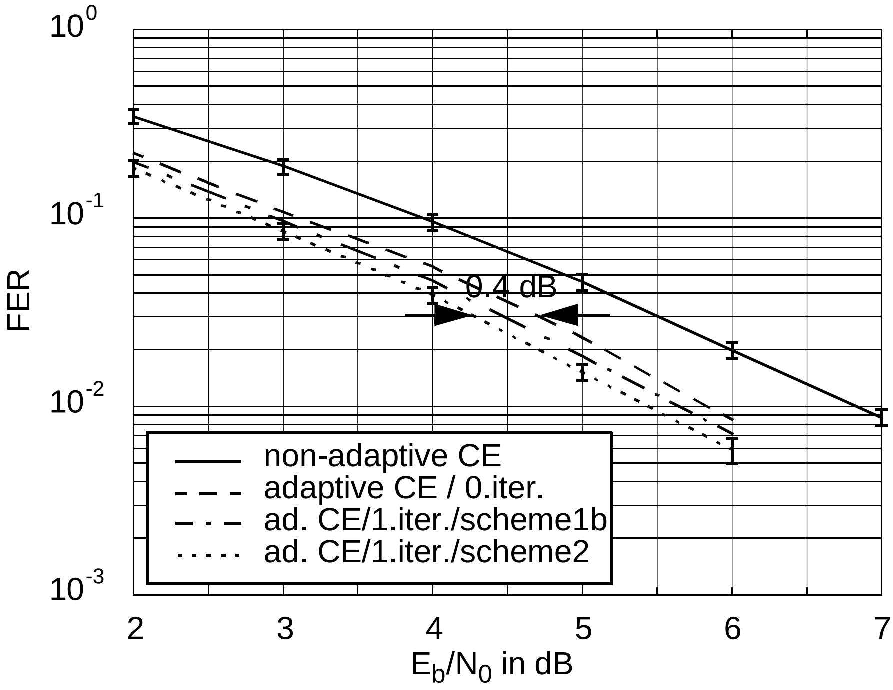


Figure 5.18. FER for RA250 (no FH) with non-adaptive CE and with adaptive CE.

6 Conclusions and outlook

In the following some of the contributions of this work are summarized and open areas of research are given.

Since the introduction of the principle of turbo-detection several research papers on this topic have been published. However, its applicability to existing mobile communication systems has not been treated. The goal of this research was to examine the benefits of turbo-detection for GSM-systems. Here, existing and future services of GSM have been treated. Subjects of the examination have been various schemes for interblock interleaving, GMSK modulated services, iterative decoding of convolutional and CRC codes, 8-PSK modulated services (EDGE), turbo-detection for serially concatenated convolutional codes and adaptive channel parameter estimation. The most important results are summarized in the following:

1. Interblock interleaving:
   In order to apply turbo-detection to interblock interleaved services, the classical turbo-detection scheme has been modified. The impact of this interleaving on turbo-detection has been investigated using the example of the fullrate speech channel (TCH/FS). Several detection schemes have been proposed and compared to the original scheme according to the error performance, the computational complexity, the delay, and the required memory. The performance has been compared for time-invariant channels significantly distorted with ISI. It was shown that by introducing an additional delay and by increasing the computational effort, the performance of the original scheme can be approached. It also turned out that without any additional delay, when compared to conventional detection, large iteration gains are still obtained.

2. GMSK modulated signals:
   The performance of selected turbo-detection schemes for interblock-interleaving has been investigated for transmission channels comprising a GMSK modulator and a time-varying multipath channel. Despite the gains that have been observed for the time-invariant channels distorted with severe ISI, only small iteration gains have been observed after the decoder. Turbo-detection improves the BER and FER after the decoder only slightly. The iteration gains after the equalizer amount to more than 2 dB for typical GSM environments. It has been shown that the reason for this is the orthogonal ISI introduced by the GMSK modulator. Only for transmission channels with significant delay spreads iteration gains can be observed. E.g. for delay-diversity transmission, iteration gains of about 0.7 dB have been observed.

3. GPRS services:
   In GPRS services, in addition to the convolutional code used for error correction, a CRC code is exploited for error detection. The CRC coder has been incorporated in the turbo-loop. The iterative decoding, utilizing SISO decoder for both codes, has

not shown any improvement of the error performance. Compared to hard-input CRC decoding, gains of about 0.5 dB have been observed for a typical fading environments. These gains again have to be sacrificed to guarantee the error detection capabilities. Neither soft-input error detection nor turbo-detection including CRC decoding enhance the performance of GPRS.

4. 8-PSK modulated EDGE services:

   To support high data rates in the near future, higher order modulation techniques will be used in GSM systems, i.e. 8-PSK modulation with a linearised GMSK pulse form filter of a bandwidth-time product of 0.3. This modulation technique introduces more ISI than the binary GMSK modulation.

   First, turbo-detection has been applied to enhanced packet switched services, i.e. EGPRS. It has been shown that the frame erasure rate can be improved for both the noise-limited and the interference-limited scenarios. Depending on the coding scheme and environment, gains of up to 2 dB have been observed.

   Depending on the signal-to-noise distribution in the network, the throughput of the system, and hence, the spectral efficiency can be improved by 10 to 30% for the noise limited scenario. For the interference limited case, the gains range from 5 to 12%.

   Second, turbo-detection has been applied to serially, concatenated convolutional codes (SCCC) that have been also proposed for enhanced circuit switched data services, i.e. ECSD. By incorporating the turbo-detection in the iterative decoding process, the iteration gains are about 1 to 2 dB higher than if only turbo-decoding is utilized.

5. Adaptive equalization:

   An adaptive MAX-Log-MAP equalizer has been combined with a modified LMS algorithm has been incorporated in the turbo-loop. Since the channel tracking of the estimator is based on the equalizer decisions, the estimation benefits from the improved equalizer performance that, e.g., occurs for GMSK modulated GSM services.

   The simulation results have shown that the new adaptive equalizer is able to nearly compensate for the time-variations of the transmission channels. However, the error rates after the decoder can be only slightly enhanced from iteration to iteration. The reason for this is that the adaptive channel tracking is mainly improved for data which is already good, and hence, cannot be further improved.

From the contributions of this work further open areas of research emerge:

- The complexity of the Max-Log-MAP equalizer for higher-order modulation techniques is very high for the GSM transmission channels. Hence, currently it is not advisable to use such an algorithm in a mobile station. The power consumption would be too high, this algorithm can be only utilized in a base station. For the 8-PSK modulated services other suboptimum equalizers, e.g. decision feedback or reduced state equalizers, should be used. In further work the performance of turbo-detection using these equalizers maybe examined.
- As already mentioned, the focus of this work has been to examine the applicability of turbo-detection to GSM-services. Since for turbo-detection the existing standard does not have to be changed, no attention was paid to principles not included in the standard. Further work maybe examine other coding and modulation techniques in combination with turbo-detection. Examples for this are space-time codes, coded modulation, other higher-order modulation techniques, etc. These examinations maybe performed with respect to a consideration for further GSM-services.

Appendix

A Log-MAP / Max-Log-MAP algorithm for equalization and decoding

The ISI channel, including the transmit filters, the mobile radio channel, and the receive filters can be regarded as a time-varying convolutional code with complex-valued coded symbols. Hence, the same algorithms can be applied for both the ISI equalization and the decoding of the convolutional code. In the following the Log-MAP algorithm and the Max-Log-MAP algorithm for equalization and for decoding are presented.

In [BCJ74], Bahl, *et al.* introduced an optimum decoding algorithm, i.e., the symbol-by-symbol MAP or the BCJR algorithm, which minimizes the bit error probability and also gives the a posteriori values of the bits. The calculations can be simplified and numerical stability can be improved by performing the calculations in the logarithmic domain, resulting in the Log-MAP algorithm [RHV95]. The Log-MAP algorithm is still optimum with respect to bit error probability and generates the same a posteriori values as the symbol-by-symbol MAP algorithm. A simplification in computational complexity can be obtained by using the Max-Log-MAP algorithm. This algorithm is suboptimal and can be directly derived from the Log-MAP algorithm.

Depending on whether the Log-MAP decoder has to decode the channel or the convolutional code, different input and output sequences are passed to or from the decoder, see Fig. 1. On equalization the Log-MAP algorithm receives the channel values **y** and the a priori values L( ) of the bit sequence **x**. From the equalizer point of view the channel values **y** denote the “code bits”; and the sequence **x,** the “information bits”. For equalization only one output sequence, i.e. the a posteriori values L( ) of the “information bits”, is generated. On decoding of the convolutional code, the Log-MAP algorithm receives two input sequences: the a priori information L( ) of the code bits **c** and the a priori information L( ) of the information bits **u**.

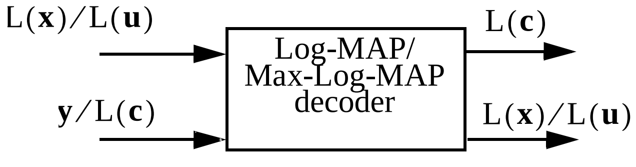

$L(\mathbf{x})/L(\mathbf{u})$
$\mathbf{y}/L(\mathbf{c})$
Log-MAP/
Max-Log-MAP
decoder
$L(\mathbf{c})$
$L(\mathbf{x})/L(\mathbf{u})$

Figure A.1. Log-MAP / Max-Log-MAP decoder.

As mentioned above the input sequences **I** are different for the decoder of the convolutional code and of the channel:

$$\mathbf{I} = \begin{cases} \mathbf{y}, \mathrm{L}(\mathbf{x}) & \text{for equalization,} \\ \mathrm{L}(\mathbf{c}), \mathrm{L}(\mathbf{u}) & \text{for decoding.} \end{cases} \tag{A.1}$$

Similar to the input sequences, the output bit O_i is defined as follows:

$$O_i = \begin{cases} \mathrm{L}(\hat{x}_i) & \text{for equalization,} \\ \mathrm{L}(\hat{c}_i^{(\mu)}), \mathrm{L}(\hat{u}_i) & \text{for decoding.} \end{cases} \tag{A.2}$$

The Log-MAP algorithm computes the a posteriori values $L(O_i)$ of the output bit O_i:

$$\mathrm{L}(O_i) = \ln \frac{\mathrm{P}(O_i = 0 | \mathbf{I})}{\mathrm{P}(O_i = 1 | \mathbf{I})}. \tag{A.3}$$

Therefore, all probabilities of the branches corresponding to the same value of the output quantity O_i are added and compared:

$$\mathrm{L}(\hat{O}_i) = \ln \frac{\sum\limits_{\{(s', s) | O_i = 0\}} \mathrm{P}(S_i = s', S_{i+1} = s, \mathbf{I})}{\sum\limits_{\{(s', s) | O_i = 1\}} \mathrm{P}(S_i = s', S_{i+1} = s, \mathbf{I})}. \tag{A.4}$$

For the calculation of the a posteriori values three quantities are important:

- The forward probability $\alpha_i(s) = \mathrm{P}(S_i = s, \mathbf{I}_1^i)$ is the joint probability of state s at time i and the input sequence from time 1 to i.
- The backward probability $\beta_i(s) = \mathrm{P}(\mathbf{I}_{i+1}^N | S_i = s)$ is the probability of the input sequence from time i+1 to N under the condition of state s at time i, where N is the size of the block to be either decoded or equalized.
- The branch transition probability $\gamma_i(s', s) = \mathrm{P}(S_i = s, \mathbf{I}_i | S_{i-1} = s')$ is the probability of state s at time i and the input value $\mathbf{I}_i$ under the condition of state s’ at time i-1. The branch transition probabilities can be distinguished between those corresponding to 0-transitions and 1-transitions: $\gamma_{i,0}(s', s)$ and $\gamma_{i,1}(s', s)$.

These three quantities are depicted in a trellis diagram in Fig. 1.

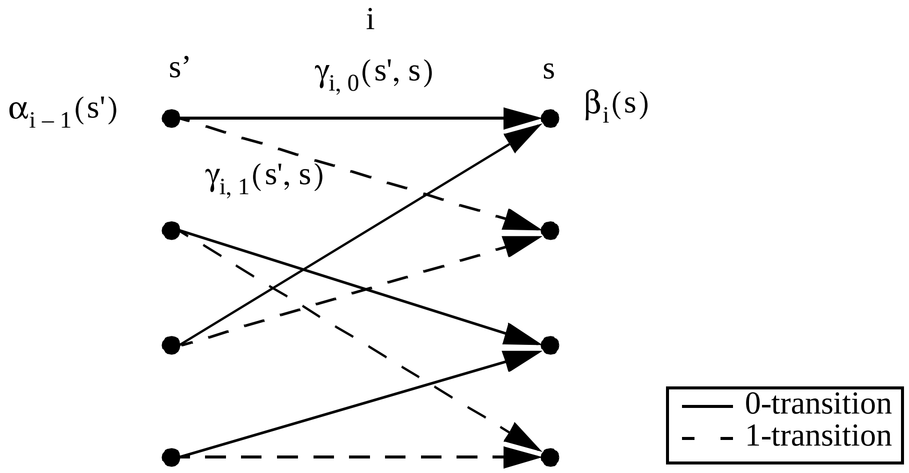


Figure A.1. Trellis diagram.

The calculations of the Log-MAP algorithm are performed in the logarithmic domain and, hence, the logarithms of the above mentioned probabilities are considered.

The logarithms of the transition probabilities are calculated differently on equalization and on decoding. The calculations on equalization are as follows:

$$\Gamma_{i,0}(s',s) = \ln\gamma_{i,0}(s',s) = L(\hat{x}_i) - \frac{\left|y_i - \hat{c}_{s',0}\right|^2}{2\hat{\sigma}^2} \text{ , and,} \qquad \text{(A.5)}$$

$$\Gamma_{i,1}(s',s) = \ln\gamma_{i,1}(s',s) = \frac{\left|y_i - \hat{c}_{s',1}\right|^2}{2\hat{\sigma}^2} \qquad \text{(A.6)}$$

On decoding, the equations are similar:

$$\Gamma_{i,0}(s',s) = \ln\gamma_{i,0}(s',s) = L(\hat{u}_i) + \sum_{\mu=1}^{n_o}(1 - c_{s',0}^{(\mu)})L(\hat{c}_i^{(\mu)}) \text{ , and,} \qquad \text{(A.7)}$$

$$\Gamma_{i,1}(s',s) = \ln\gamma_{i,1}(s',s) = \sum_{\mu=1}^{n_o} c_{s',0}^{(\mu)}L(\hat{c}_i^{(\mu)}) \text{ .} \qquad \text{(A.8)}$$

The logarithm of the forward probability is obtained by the forward recursion

$$\ln\alpha_i(s) = \ln\sum_{s'}\exp(\ln\alpha_{i-1}(s') + \ln\gamma_i(s',s)), \qquad \text{(A.9)}$$

while the logarithm of the backward probability is obtained by the backward recursion

$$\ln\beta_i(s) = \ln\sum_{s'}\exp(\ln\beta_{i+1}(s') + \ln\gamma_{i+1}(s, s')) . \tag{A.10}$$

The output quantity O_i is then calculated by:

$$L(\hat{O}_i) = \ln \sum_{\{(s', s)|O_i = 0\}} \exp(\ln\alpha_{i-1}(s') + \ln\gamma_i(s', s) + \ln\beta_i(s)) - \tag{A.11}$$
$$-\ln \sum_{\{(s', s)|O_i = 1\}} \exp(\ln\alpha_{i-1}(s') + \ln\gamma_i(s', s) + \ln\beta_i(s)) .$$

The equations (9), (10), and (11) can be evaluated using the Jacobian logarithm

$$\ln\{\exp(\delta_1) + \exp(\delta_2)\} = \max\{\delta_1, \delta_2\} + f_c(|\delta_1 - \delta_2|) , \tag{A.12}$$

where

$$f_c(|\delta_1 - \delta_2|) = \ln(1 + \exp(-|\delta_1 - \delta_2|)) \tag{A.13}$$

is a correction function that can be implemented using a look-up table.

If the convolutional code or the channel are non-recursive, the a posteriori value calculation of $L(x_i)$ and $L(u_i)$ simplify to:

$$L(\hat{O}_i) = \ln \sum_{\{s|O_i = 0\}} \exp(\ln\alpha_i(s) + \ln\beta_i(s)) - \tag{A.14}$$
$$-\ln \sum_{\{s|O_i = 1\}} \exp(\ln\alpha_i(s) + \ln\beta_i(s)) .$$

For turbo-detection this simplification only reduces the computational complexity for the equalizer. This is due to the fact that the L-values $L(\mathbf{c})$ of the code bits **c** have to be calculated in the decoder.

The Max-Log-MAP algorithm can be deduced from the Log-MAP algorithm by an approximation. The correction term in (12) is disregarded in the calculations of (9), (10), (11), and (14). The algorithm now consists of two VA, one running forward through the trellis to calculate the logarithm of the α’s and one running backward to compute the logarithm of the β's:

$$\ln\alpha_i(s) = \max_{s'}\{\ln\alpha_{i-1}(s') + \ln\gamma_i(s', s)\}, \text{ and,} \tag{A.15}$$

$$\ln\beta_i(s) = \max_{s'}\{\ln\beta_{i+1}(s') + \ln\gamma_{i+1}(s, s')\} . \tag{A.16}$$

The soft-outputs are calculated by

$$L(\hat{\ }) = \max_{\{\ |\ \}} \{ \ \} - \qquad \text{(A.17)}$$
$$- \max_{\{\ |\ \}} \ .$$

If the convolutional code or the channel are non-recursive, the values $L(x_i)$ and $L(\ddot{u}_i)$ can also be calculated with

$$L(\hat{O}_i) = \max_{\{s|O_i = 0\}} \{\ln\alpha_i(s) + \ln\beta_i(s)\} - \qquad \text{(A.18)}$$
$$- \max_{\{s|O_i = 1\}} \{\ln\alpha_i(s) + \ln\beta_i(s)\}.$$

B The rotator LMS-algorithm

In communication systems the transmission channel can be regarded as serial concatenation of the modulator, the waveform channel, i.e. the mobile radio channel in mobile communication systems, and the receiver front end plus receive filter. The delay spread of most GSM-environments is short, e.g. most of the energy is received within one symbol duration T. Since the system is not able to dissolve paths arriving within a delay smaller than T, the fading environment is mainly represented by one fading tap in the receiver. This special channel characteristics can be taken into account by introducing additional constraints to the LMS algorithm [Rad96].

The structured channel to be estimated can be approximated by the following model:

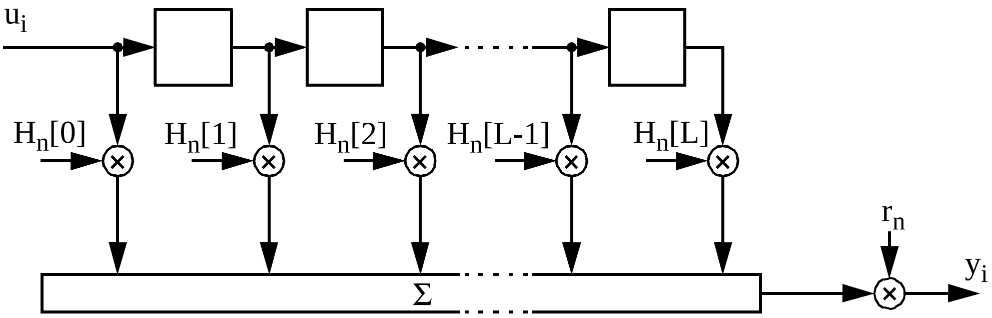


Figure B.1. Time discrete ISI channel model.

The time-variant channel impulse response at time instant n is split into two parts: the partial channel impulse response vector $\mathbf{H}_n$ and the rotator r_n. The rotator r_n denotes a common factor for all the components of the channel impulse response. This common factor represents one fading tap. The estimated channel impulse response $\mathbf{h}_n$ is given by

$$\mathbf{h}_n = r_n \mathbf{H}_n. \tag{B.1}$$

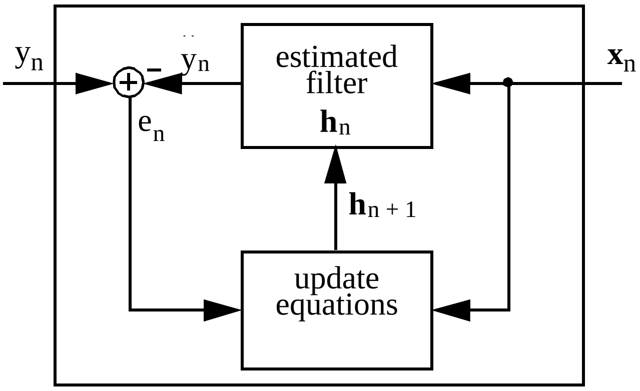


Figure B.2. LMS-algorithm.

The structure of the LMS algorithm is depicted in Fig. 2. The last L+1 transmitted symbols $\mathbf{x}_n=\{x_n, x_{n-1}, ..., x_{n-L}\}^T$ are multiplied with an estimated channel impulse response $\hat{\mathbf{h}}_n = \{\hat{h}_n^{(0)}, \hat{h}_n^{(1)}, \ldots, \hat{h}_n^{(L)}\}^T$, resulting in the estimated channel value

$$\hat{y}_n = \hat{\mathbf{h}}_n^T \mathbf{x}_n . \tag{B.2}$$

This value $\hat{y}_n$ is then compared with the received value y_n and the estimation error e_n is obtained:

$$e_n = y_n - \hat{y}_n . \tag{B.3}$$

Now, all values for the update equations are available: the error value e_n, the vector $\mathbf{x}_n$, and the actual estimated vector $\mathbf{h}_n$ of the channel impulse response. With these values the new updated channel impulse response $\mathbf{h}_{n+1}$ can be calculated.

In the following new update equations are deduced taking into account the new constraints of the channel. These update equations denote the so-called rotator LMS algorithm. For the deduction of the original LMS algorithm see [Hay96]. Note that the index n representing the time instant is not given in the following deduction for clarity.

Considering the channel structure given in (1) the estimation error e is given by

$$e = y - \hat{\mathbf{h}}^T \mathbf{x} = y - \hat{r}\hat{\mathbf{H}}^T \mathbf{x} . \tag{B.4}$$

With (4) the cost function, i.e. the mean square error, can be calculated as following:

$$C = ee^* = (y - r\mathbf{H}^T\mathbf{x})(y^* - r^*\mathbf{H}^H\mathbf{x}^*) . \tag{B.5}$$

Using the method of the steepest descent, the cost function C has to be derived. Here, an extended vector $\mathbf{h}_e$ is used in order to incorporate the channel structure. The extended LMS vector $\mathbf{h}_e$ is a vector of L+2 scalars:

$$\mathbf{h}_e = \begin{bmatrix} \mathbf{H} \\ r \end{bmatrix} . \tag{B.6}$$

The derivation $\frac{dC}{d\mathbf{h}_e}$ can be distinguished in two equations:

$$2\frac{\partial C}{\partial \mathbf{H}^*} = e\frac{\partial}{\partial \mathbf{H}^*}(e^*) + e^*\frac{\partial}{\partial \mathbf{H}^*}(e) = -(y - r\mathbf{H}^T\mathbf{x})r^*\mathbf{x}^* = -\mathbf{x}^*r^*\epsilon\text{, and,} \tag{B.7}$$

$$2\frac{\partial C}{\partial r^*} = e\frac{\partial}{\partial r^*}(e^*) + e^*\frac{\partial}{\partial r^*}(e) = -(y - r\mathbf{H}^T\mathbf{x})\mathbf{H}^*\mathbf{x}^* = -\mathbf{H}^*\mathbf{x}^*e \quad . \tag{B.8}$$

From here on it is again useful to use the index n for the time instant. The equations (7) and (8) can now be inserted in the steepest descent equation

$$\mathbf{h}_{e,\,n+1} = \mathbf{h}_{e,\,n} + \mu\left(-\frac{dC}{d\mathbf{h}_{e,\,n}}\right) . \tag{B.9}$$

The steepest descent equation can be split in two parts:

$$\mathbf{H}_{n+1} = \mathbf{H}_n + \mu_h\mathbf{x}^*r^*{}_n e \text{ and} \tag{B.10}$$

$$r_{n+1} = r_n + \mu_r\mathbf{H}_n^*\mathbf{x}^*e . \tag{B.11}$$

In contrast to the original LMS algorithm two update equations are obtained. For an appropriate tracking behaviour, the adaptation constants μ_h and μ_r have to be chosen carefully.

On the one hand, the algorithm is able to follow the time variations faster if the constants are large. On the other hand, if the constants are chosen too large, the algorithm becomes unstable.

The advantage of the rotator LMS algorithm is that the constant μ_r can be chosen larger in order to track the fast time-variations of the fading tap. The constant μ_h is chosen as low as if the original LMS algorithm would be applied. For the exact setting of the constants, simulations can be carried out. However, there is no optimal parameter set for all environments and a compromise has to be found for the settings that works for all environments. Note that for the simulations of Chapter 5 it showed that the following values are well suited: μ_r=0.05, μ_h=0.0125.

C Soft-In Error-Detection

In GSM systems, blockcodes are used to detect erroneous transmitted frames at the receiver and to guarantee low residual block error rates. The use of CRC codes for error detection comprises two main advantages:

- Low residual bit error rates can be guaranteed with low redundancy.
- On decoding, a low complexity hard-input decoder can be utilized.

In order to improve the error performance of the block code, a SISO decoder can be utilized. However, the error detection capability is sacrificed if the SISO decoder is exclusively utilized to correct errors. To maintain the error detection performance, a new decoding strategy that enables also the detection of erroneous blocks is exploited.

In the following, the SISO algorithm of Section 2.2.2 is used to both, to correct errors more effectively than with conventional hard-decision decoding as well as to detect those frames that are still not correct. If the L-values are large, it can be assumed that the decisions are correct. If the L-values are small, an error is very likely and the frame is rejected.

This principle is illustrated at the example of a (28,23)-shortened cyclic block code with the generator polynomial $g(\quad) = 1 + \hat{\ }\ + D^5$ is considered. Note that this code is not used in any GPRS coding scheme. However, it is well suited to explain the new error detection principle. The minimum trellis for this code has $2^5 = 32$ states. In Fig. 1 the FER for several decoding strategies are depicted. The worst error rate is obtained with hard-decision error detection using syndrome calculation. The lowest error rate is achieved if the errors are corrected with the SISO decoder. A performance between these two is obtained with the new decoding strategy. Here, the SISO decoder corrects the received sequence. After correction the soft-output values are examined. The BFI is set to zero, i.e. the block is considered to be correctly transmitted if the L-values are larger than a certain limit, otherwise the BFI is set to one and rejected. The detection rule used for simulation is:

$$\mathrm{BFI} = \begin{cases} 0, \text{ if } \min\{|\quad \hat{\ }\quad|\} \geq 2\mathrm{erfc}(\qquad) \\ 1, \text{ if } \min\{|\quad \hat{\ }\quad|\} < 2\mathrm{erfc}(\qquad) \end{cases}. \tag{C.1}$$

To be able to evaluate the performances of these three schemes, the number of undetected erroneous blocks, i.e. the RFER has to be considered. In Fig. 2 the RFER of the three decoding strategies is presented. Although the received sequence is corrected a large number of erroneous frames are still passed to the data sink. The hard-decision syndrome error detector and the new BCJR correction/detection strategy have approximately the same RFER performance.

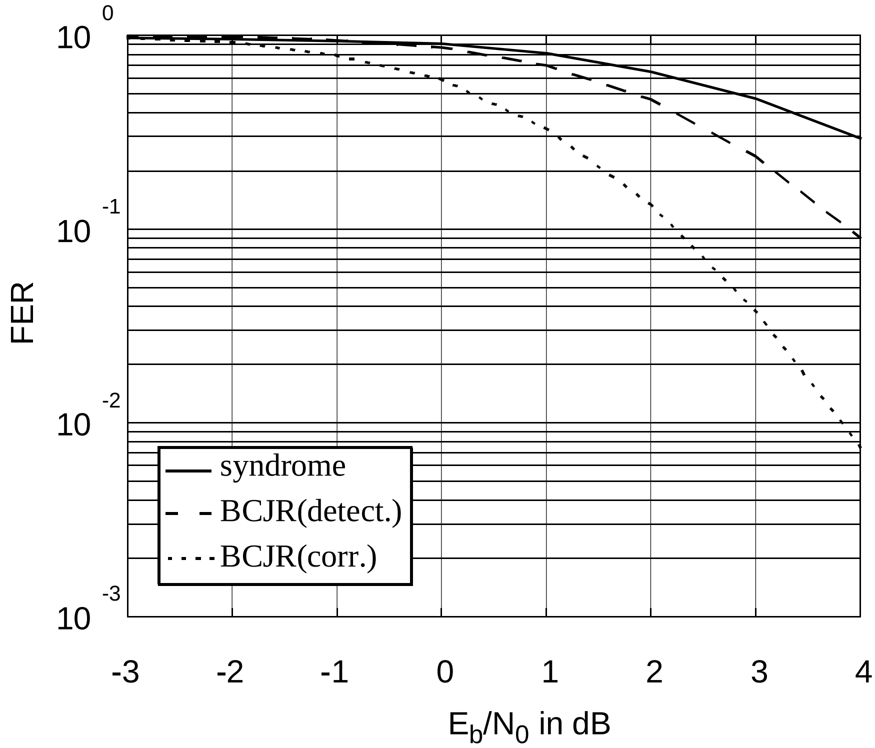


Figure C.1. FER for (28,23)-shortened cyclic block code with syndrome error detection, SISO error correction and SISO error correction and detection.

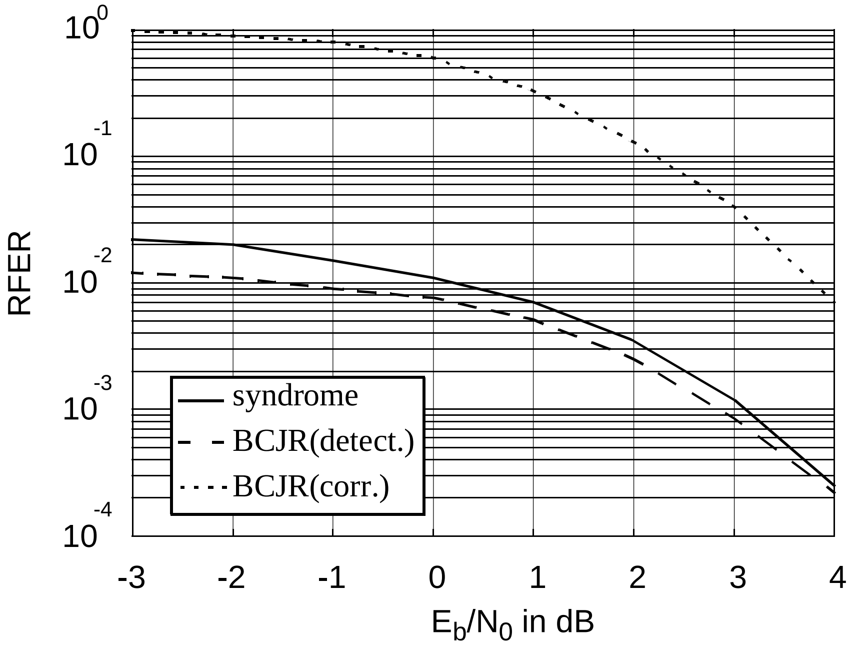


Figure C.2. RFER for (28,23)-shortened cyclic block code with syndrome error detection, BCJR error correction and BCJR error correction and detection.

With this new correction/detection strategy a large gain can be achieved. With approximately the same error detection capability more erroneous blocks can be corrected. For ARQ transmission this means an improvement in throughput because less frames have to be retransmitted.

D List of frequently used symbols and abbreviations

List of abbreviations

ACI	adjacent channel interference
APP	a posteriori probability
ARQ	automatic repeat request
AWGN	additive white Gaussian noise
BCH	Bose Chaudhuri Hoquenghem
BPSK	binary phase shift-keying
BER	bit error rate
BFI	bad frame indicator
BS	base station
BT	bandwidth-time product
cdf	cumulated probability density function
CE	channel estimation
C/I	carrier-to co-channel-interferer ratio
CRC	cyclic redundancy check
CRE	channel re-estimation
CS	coding scheme (used in GPRS transmission)
D-AMPS	digital advance phone service
ECSD	enhanced circuit switched data
EDGE	enhanced data services for GSM evolution
EGPRS	enhanced general packet radio services
ETSI	European Telecommunications Standards Institute
FER	frame erasure rate

FH	frequency hopping
GF(2)	Galois field 2
GMSK	Gaussian minimum shift keying
GPRS	general packet radio services
GSM	Global System for Mobile Communications, previously: “Groupe Spéciale Mobile”
HSCSD	high speed circuit switched data
HT	hilly terrain
ISI	intersymbol interference
LMS	least-mean-square
MAP	maximum a posteriori
ML	maximum likelihood
MLSE	maximum likelihood sequence estimation
PCCC	parallel concatenated convolutional code
PDC	pacific digital cellular
pdf	probability density function
PSK	phase shift keying
RA	rural area
RBER	residual bit error rate
RFER	residual frame erasure rate
RLS	recursive least-squares
RS	Reed Solomon
SCCC	serially concatenated convolutional code
SISO	soft-in/soft-out
SOVA	soft-output Viterbi algorithm

TCH/FS	traffic channel/full-rate speech
TDMA	time-division multiple access
TU	typical urban
USDC	United States digital cellular
VA	Viterbi algorithm
WSSUS	wide-sense stationary uncorrelated scattering

List of symbols

γ_{db}	local mean of signal-to-noise ratio
γ	transition probability
Γ	logarithm of transition probability
Π	interleaver
Π^{-1}	deinterleaver
ρ	throughput of automatic repeat request scheme
σ_{dB}	standard deviation of local mean of signal-to-noise ratio
σ	standard deviation of white Gaussian noise
C	linear block code
$\mathbf{c}$	vector representation of code sequence
$\mathbf{c}$	estimated code sequence
c(D)	polynomial representation of code bit sequence
$c^{(i)}(D)$	ith code sequence of a convolutional code
c_i	ith code bit
$\hat{c}_i$	ith estimated code bit
c(t)	complex lowpass signal

C_f	reference detection complexity
$C_{f,e}$	reference equalization complexity
$C_{f,d}$	reference decoding complexity
d_i	i^{th} symbol
d_{min}	the minimum Hamming distance of a code
$d[n]$	training sequence
$\mathbf{e}$	error vector
E_b	energy per modulated bit
f_0	carrier frequency
f_D	Doppler frequency
$F[D]^i$	class of polynomials of grade i
$F[D]$	ring of all polynomials
$F_n[D]$	ring of polynomial of grade smaller or equal to n
F^{n}	n-dimensional Galois field two
$g(D)$	generator polynomial of cyclic code
$g^{(i)}(D)$	generator polynomial i of a convolutional code
$\mathbf{G}$	generator matrix of a block code
$\mathbf{G}(D)$	generator matrix of a convolutional code
H	test matrix
$h(D)$	test polynomial
$h(t)$	time-continuous channel impulse response
$\mathbf{h}$	vector of time-discrete channel impulse response
$\mathbf{h}_n$	vector of time-discrete, time-varying channel impulse response at time n
$h[n]$	time-discrete channel impulse response

h_i	i^{th} tap of time-discrete channel impulse response
$h_{TX}(t)$	channel impulse response of transceiver filter
$\mathbf{I}_k$	k-dimensional identity matrix of a systematic code
$L(.)$	L-value, log-likelihood ratio
$L_d(.)$	symbol L-value
$L^D(.)$	a posteriori L-value at output of decoder
$L_e^D(.)$	extrinsic L-value at output of decoder
$L^E(.)$	a posteriori L-value at the output of equalizer
$L_*^E(.)$	extrinsic L-value and channel information at the output of equalizer
L	memory length of transmission channel
m	memory length of convolutional code
M	size of modulation alphabet
m_{dB}	mean of signal-to-noise ratio in a cell
n_0	number of output bits per input bit for a convolutional code
N_0	noise density
$\mathbf{p}$	parity sequence of systematic block code
$\mathbf{P}$	parity submatrix
$P(.)$	probability of
$\mathbf{r}$	received sequence at the block decoder
R	code rate
$\mathbf{s}$	syndrome of a block code
s	state of the trellis
S_i	state of the trellis at stage i
$\mathbf{t}$	tail bit sequence

T	symbol duration
T_s	sampling period
$\mathbf{u}$	information bit sequence
$\hat{\mathbf{u}}$	estimated information sequence
u(D)	polynomial representation of information sequence
u_i	i^{th} information bit
$\hat{u}_i$	i^{th} information bit
$w_h(\mathbf{c})$	Hamming weight of the code word **c**
$\mathbf{x}$	interleaved code bit sequence
$\hat{\mathbf{x}}$	estimated interleaved code bit sequence
x_i	i^{th} bit of interleaved sequence
$\hat{x}_i$	i^{th} bit of estimated interleaved sequence
$\mathbf{y}$	sampled output values of receive filter
y[n]	sampled output values of receive filter

Bibliography

[AAS86] J. B. Anderson, T. Aulin, and C.-E. Sundberg, *Digital Phase Modulation.* New York, USA: Plenum, 1986.

[AnM91] J. B. Anderson and S. Mohan, *Source and Channel Coding - An Algorithmic Approach,* 1st edition. Normwell, MA, USA: Kluwer Academic Publishers, 1991.

[BaF98a] V. Franz and G. Bauch, “Iterative channel estimation for “turbo”-detection,” in *Proceedings of ITG-Fachtagung ‘Codierung für Quelle, Kanal und Übertragung’,* Aachen, Germany, March 1998, pp. 149-154.

[BaF98b] G. Bauch and V. Franz, “A comparison of soft-in/soft-out algorithms for “turbo”-detection”, in *Proceedings of International Conference on Telecommunications*, Porto Carras, Greece, June 1998, pp. 259-264.

[BaF98c] G. Bauch and V. Franz, “Iterative equalization and decoding for the GSM-system,” in *Proceedings of IEEE Vehicular Technology Conference*, Ottawa, Canada, May 1998, pp. 2262-2266.

[BaF99] G. Bauch and V. Franz, “Turbo-detection for enhance data for GSM evolution,” to published in *Proceedings of IEEE Vehicular Technology Conference Fall*, Amsterdam, The Netherlands, September 1999.

[Bai90] A. Baier, “Derotation techniques in receivers for MSK-type CPM signals,” in *Proceedings of EUSIPCO’90,* Barcelona, Spain, vol.3, September 1990, pp. 1799-1802.

[BCJ74] L.R. Bahl, *et al.*, “Optimal decoding of linear codes for minimizing symbol error rate,” *IEEE Transactions on Information Theory,* vol. 20, pp. 284-287, March 1974.

[BCT98] E. Biglieri, G. Caire, and G. Taricco, “Coding for the fading channel: a survey,” in *Proceeding of 1998 NATO-ASI*, Il Ciocco, Italy, July 1998.

[BDG79] G. Battail, M. C. Decouvelaere, and P. Godlewski, “Replica Decoding,” *IEEE Transactions on Information Theory,* vol. 25, no.3, May 1979.

[BDM97a] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, “A soft-output APP module for iterative decoding of concatenated codes,” *IEEE Communications Letters*, vol. 1, no. 1, pp. 22-24, January 1997.

[BDM97b] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, “Design of serially concatenated interleaved codes,” in *IEEE International Conference on Communications*, Montreal, Canada, June 1997, vol. 2, pp. 710-714.

[BDM98a] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, “Analysis, design, and iterative decoding of double serially concatenated codes with interleavers,” *IEEE Journal on Selected Areas in Communications,* vol. 16, no. 2, pp.231-244, February 1998.

[BDM98b] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, “Serial concatenation of interleaved codes: performance analysis, design, and iterative decoding,” *IEEE Transactions on Information Theory*, vol. 44., no. 3, pp. 909-926, March 1998.

[BeG96] C. Berrou, and A. Glavieux, “Near optimum error correcting coding and decoding: turbo-codes,” *IEEE Transactions on Communications,* vol. 44, no. 10, pp. 1261-1271, October 1996.

[Bel63] P. A. Bello, “Characterization of randomly time-variant linear channels,” *IEEE Transactions on Communications,* vol. 11, pp. 360-393, December 1963.

[BeM96a] S. Benedetto, and G. Montorsi, “Unveiling turbo codes: some results on parallel concatenated coding schemes,” *IEEE Transactions on Information Theory*, vol. 42, no.2 , pp. 409-428, February 1996.

[BeM96c] S. Benedetto, and G. Montorsi, “Serial concatenation of block and convolutional codes,” *Electronic Letters*, vol. 32, no. 10, pp. 887-888, 1996.

[BeM96d] S. Benedetto, and G. Montorsi, “Iterative decoding of serially concatenated convolutional codes,” *Electronic Letters,* vol. 32, no. 13, pp. 1186-1188, August 1996.

[BeS86] Y. Beery, and J. Snyders, “Optimal soft decision block decoders based on fast Hadamard transform,” *IEEE Transactions on Information Theory,* vol. 32, no. 3, pp. 355-364, May 1986.

[BGT93] C. Berrou, A. Glavieux, and P. Thitimajshima, “Near Shannon limit error-correction and decoding: Turbo-codes (1),” in *IEEE International Conference on Communications (ICC),* ort, land, May 1993, p. 1064-1070.

[BKH97] G. Bauch, H. Khorram, and J. Hagenauer, “Iterative equalization and decoding in mobile communication systems,” in *The Second European Personal Mobile communications Conference (2. EPMCC’97) together with 3. ITG-Fachtagung ‘Mobile Kommunikation’,* Bonn, Germany, October 1997, pp. 307-312.

[Bla90] R. E. Blahut, *Digital Transmission of Information*. Reading, MA, USA: Addison-Wesley Publishing Company, 1990.

[Bos98] M. Bossert, *Kanalcodierung,* 2nd edition. Stuttgart, Germany: B.G. Teubner, 1998.

[Cha72] D. Chase, “A class of algorithms for decoding block codes with channel measurement information,” *IEEE Transactions on Information Theory,* vol. 18, no. 1, January 1972.

[CJL94] C. Carneheim, S.-O. Jonsson, M. Ljungberg, M. Madfors, and J. Naslund, “FH-GSM frequency hopping GSM,” in *Proceeding of IEEE Vehicular Technology Conference,* Stockholm, Sweden, June 1994, vol. 2, pp. 1155-1159.

[COS89] COST 207, *Digital Land Mobile Radio Communication - Final Report*. Luxembourg: Office for Official Publications of the European Communities, 1989.

[DJB95] C. Douillard, M. Jézéquel, C. Berrou, A. Picart, P. Didier, and A. Glavieux, “Iterative correction of intersymbol interference,” *European Transactions on Telecommunications,* vol. 6, pp. 507-511, September - October 1995.

[Eli54] P. Elias, “Coding for noisy channels,” *IRE Conv. Rec.,* Part 4, pp. 37-47, 1955.

[ETS96] ETSI, “Digital cellular telecommunications system; Technical performance objectives,” ETR GSM 03.05, version 5.0.0, November 1996.

[ETS97] ETSI, “Digital cellular telecommunications system (Phase 2+); Modulation,” ETS GSM 05.04, version 5.0.1, May 1997.

[ETS98a] ETSI, “EDGE - evaluation of 8-PSK,” *Tdoc SMG2 WPB 108/98,* Kuusamo, Finland, April 1998.

[ETS98b] ETSI, “Digital cellular telecommunications system (Phase 2+); Channel coding,” ETS GSM 05.03, version 5.5.0, May 1998.

[ETS98c] ETSI, “Digital cellular telecommunications system (Phase 2+); Radio transmission and reception,” ETS GSM 05.05, version 5.9.0, July 1998.

[ETS98d] ETSI, “Digital cellular telecommunications system (Phase 2+); Physical layer on the radio path; General Description,” ETS GSM 05.01, version 5.4.0, April 1998.

[ETS98e] ETSI, “EDGE - Evaluation of 8-PSK,” Tdoc SMG2 WPB#4 108/98, April 1998.

[ETS98f] ETSI, “Serially Concatenated Codes for EDGE channel coding,” Tdoc SMG2 WPB 057/98, August 1998.

[ETS99] ETSI, “Digital cellular telecommunications system (Phase 2+); Enhanced data rates for GSM evolution (EDGE); Project scheduling and open issues for EDGE,” PT SMG PD/GSM 10.59, version 1.13.0, May 1999.

[FoL95] M. P. Fossorier, and S. Lin, “Soft-decision decoding of linear block codes based on ordered statistics,” *IEEE Transactions on Information Theory,* vol. 41, no. 5, pp. 1379-1396, September 1995.

[For66a] G. D. Forney, *Concatenated Codes*. Cambridge, MA: M.I.T. Press, 1966.

[For66b] G. D. Forney, “Generalized Minimum Distance Decoding,” *IEEE Transactions on Information Theory*, vol. 12, no. 2, pp. 125-131, April 1966.

[For72] G. D. Forney, Jr., “Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference,” *IEEE Transactions on Information Theory*, vol. 18, no.3, pp. 363-378, May 1972.

[FrA97] V. Franz and J. B. Anderson, “Reduced-search BCJR algorithms,” in *IEEE International Symposium on Information Theory (ISIT)*, Ulm, Germany, June/July 1997, p. 230.

[FrA98] V. Franz and J. B. Anderson, “Concatenated decoding with a reduced-search BCJR algorithm,” *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 2, pp. 186-195, February 1998.

[Fra96] V. Franz, “The Soft-output M-algorithm,” Diplomarbeit, Lehrstuhl für Nachrichtentechnik, TU München, June 1996.

[Fri95] B. Friedrichs, *Kanalcodierung - Grundlagen und Anwendungen in modernen Kommunikationssystemen*. Berlin, Germany: Springer-Verlag, 1995.

[GLL97] A. Glavieux, C. Laot, and J. Labat, “Turbo equalization over a frequency selective channel,” in *Proceedings of International Symposium on Turbo Codes*, Brest, France, September 1997, pp. 96-102.

[HaA96] Ulf Hansson, “Soft information transfer for sequence detection with concatenated receivers,” *IEEE Transactions on Communications*, vol. 44, no. 9, pp. 1086-1095, September 1996.

[Hae98] J. Hämäläinen, “The GPRS specification,” in *Proceedings of GSM World Congress’98*, Cannes, France, February 1998.

[Hag92] J. Hagenauer, “Soft-in/soft-out the benefits of using soft values in all stages of digital receivers,” in *Proc. of the 3rd Int. Workshop on Digital Signal Processing*, ESTEC, Noordwijk, The Netherlands, September 1992.

[Hag94] J. Hagenauer, “Soft is better than hard,” in R. E. Blahut and D. J. Costello, *Coomunications and Cryptography - Two Sides of One Tapestry*. Kluwer Academic Publisher, 1994.

[Hag95] J. Hagenauer, “Source-controlled channel decoding,” *IEEE Transactions on Communications*, vol. 43, no. 9, pp. 2449-2457, September 1995.

[Hag97] J. Hagenauer, “The turbo principle: Tutorial introduction and state of the art,” in *Proceedings of International Symposium on turbo-codes*, Brest, France, September 1997.

[HNB97] J. Hagenauer, H. Nickl and F. Burkert, “Approaching Shannon’s capacity limit by 0.27 dB using Hamming-Codes in a ‘turbo’-decoding scheme,” in *IEEE International Symposium on Information Theory,* Ulm, Germany, June 1997, p. 12.

[HOP96] J. Hagenauer, E. Offer, and L. Papke, “Iterative decoding of binary block and convolutional codes,” *IEEE Transactions on Information. Theory*, vol. 42, no.2, pp. 429-445, March 1996.

[HaH89] J. Hagenauer and P. Höher, “A Viterbi algorithm with soft-decision outputs and its applications,” in *Proc. Globecom’89,* Dallas, Texas, November 1989, pp. 47.1.1-47.1.7.

[HaR76] C. R. Hartmann, and L. D. Rudolph, “An optimum symbol-by-symbol decoding rule for linear codes for linear codes,” *IEEE Transactions on Information Theory,* vol. 13, pp. 514-517, September 1976.

[Hay96] S. Haykin, *Adaptive Filter Theory*, 3rd edition. New Jersey, USA: Prentice Hall, 1996.

[HeJ71] J. A. Heller and I. M. Jacobs, “Viterbi decoding for satellite and space communications,” *IEEE Transactions on Communications Technology,* pp. 835-848, October 1971.

[HiR97] T. Hindelang and A. Ruscitto, “Channel decoding using residual intraframe correlation in a GSM system,” *Electronic Letters,* vol.33, no.21, pp. 1754-1755, November 1997.

[HiR98] T. Hindelang and A. Ruscitto, “Channel decoding with a priori knowledge for non-binary source symbols,” in *Proceedings of ITG-Fachtagung ‘Codierung für Quelle, Kanal und Übertragung’*, Aachen, Germany, March 1998, pp. 163-167.

[HRP94] J. Hagenauer, P. Robertson, and L. Papke, “Iterative (‘TURBO’) decoding of systematic convolutional codes with the MAP and SOVA algorithm,” *Proceedings of ITG-Fachtagung ‘Codierung für Quelle, Kanal und Übertragung’*, Munich, Germany, October 1994, pp. 21-29.

[Jak74] W. C. Jakes, Jr., *Microwave Mobile Communications*. New York, USA: John Wiley & Sons, 1974.

[JBS98] R. Jordan, M. Bossert, and G. Schnabl, “Description of woven codes with internal interleaving,” 1998.

[JoZ99] R. Johannesson and K. Zigangirov, *Fundamentals of Convolutional Coding,* 1st edition. New York, NY, USA: IEEE Press, 1999.

[JuB92] P. Jung, and P. W. Baier, “On the representation of CPM signals by linear superposition of impulses in the bandpass domain,” *IEEE Journal on Selected Areas in Communications,* vol. 10, no.8, pp. 1236-1242, October 1992.

[Kam92] K. D. Kammeyer, *Nachrichtenübertragung*. Stuttgart, Germany: B. G. Teubner, 1992.

[Kho97] H. Khorram, “Iterative Entzerrung und Kanaldecodierung bei frequenzselektiven Kanälen mit dem MAP-Algorithmus,” Diplomarbeit, Lehrstuhl für Nachrichtentechnik, TU München, January 1997.

[KNI94] T. Kaneko, T. Nishijima, H. Inazumi, and S. Hirasawa, “An efficient maximum likelihood decoding of linear block codes with algebraic decoders,” *IEEE Transactions on Information Theory*, vol. 40, pp.320-327, March 1994.

[KoB90] W. Koch and A. Baier, “Optimum and Suboptimum detection of coded data disturbed by time-varying intersymbol interference,” in *Proceedings of IEEE International Conference on Communications*, San Diego, USA, 1990, pp. 1679-1684.

[Lau86] P. A. Laurent, “Exact and approximate construction of digital phase modulations by superposition of amplitude modulated pulses (AMP),” *IEEE Transactions on Communications*, vol. 34, no. 2, pp. 150-160, February 1986.

[Lee93] W. C. Y. Lee, *Mobile communications design fundamentals*. New York, USA: John Wiley & Sons, 1993.

[LiC83] S. Lin and D. J. Costello, jr. , *Error Control Coding*. New Jersey: Prentice-Hall, 1983.

[LBB98] R. Lucas, M. Breitbach, and M. Bossert, “On iterative soft-decision decoding of linear binary block codes and product codes,” *IEEE Journal on Selected Areas in Communication*, vol. 16, no. 2, pp. 276-296, February 1998.

[LYH93] J. Lodge, R. Young, P. Hoeher, and J. Hagenauer, “Separable MAP ‘filters’ for the decoding of product and concatenated codes,” in *Proc. IEEE Int. Conf. Communications (ICC)*, Geneva, Switzerland, May 1993, pp. 1740-1745.

[MaP73] F. R. Magee and J. G. Proakis, “Adaptive maximum-likelihood sequence estimation for digital signaling in the presence of intersymbol interference,” *IEEE Transactions on Information Theory*, vol. 19, pp. 120-124, January 1973.

[MeM92] R. Mehlan and H. Meyr, “Soft output M-algorithm equalizer and trellis-coded modulation for mobile radio communication,” in *Vehicular Technology Society 42nd VTS Conference*, Denver, USA, May 1992, vol. 2, pp. 586-591.

[MoK96] W. Mohr and A. Klein, “Considerations on the applicability of wideband channel models for frequency hopping investigations,” AC090/SM/AI0/Pi/I/091/a1, FRAMES document, August 1996.

[MoP92] M. Mouly and M.-B. Pautet, *The GSM System for Mobile Communivations*. Palaiseau, France: Cell & Sys, 1992.

[NeM91] P. Newson and B. Mulgrew, “Adaptive channel identification and equalisation for GSM european digital mobile radion,” in *Proceeding of IEEE International Conferennce on Communications*, Denver, USA, June 1991, vol. 1, pp. 23-27.

[OAH97] H. Olofsson, M. Almgren, and M. Hook, “Transmitter diversity with antenna hopping for wireless communication systems,” in *Proceedings of IEEE Vehicular Technology Conference,* Phoenix, USA, May 1997, vol. 3, pp. 1743-1747.

[PDG97] A. Picart, P. Didier, and A. Glavieux, “Turbo-detection: a new approach to combat channel frequency selectivity,” in *IEEE International Conference on Communications,* Montreal, Canada, June 1997, vol. 3, pp. 1498-1502.

[Per92] Personal Digital Cellular, Japanese Telecommunication System Standard, RCR STD 27.B, 1992.

[Pra57] E. Prange, “Cyclic error-correcting codies in two symbols,” *AFCRC-TN-57,* Air Force Combridge Research Center, Cambridge, Amss., September 1957.

[Pro91] J. G. Proakis, “Adaptive equalization for TDMA digital mobile radio,” *IEEE Transactions on Vehicular Technology,* vol. 40, no. 2, May 1991.

[Pro95] J. G. Proakis, *Digital Communications*. 3rd edition, New York, USA: McGraw-Hill, 1995.

[Qur85] S. U. H. Qureshi, “Adaptive equalization,” *Proceedings of the IEEE,* vol. 73. no. 9, September 1985.

[Rad96] L. Rademacher, “An additional common factor for all tap-weights in adaptive algorithms to accelerate adaptation and minimize noise amplification,” *Patent GR96P1995EE,* German Patent Office, May 1996.

[Rap96] T. S. Rappaport, *Wireless Communications*. New Jersey, USA: Prentice Hall, 1996.

[RaZ97] D. Raphaeli and Y. Zarai, “Combined turbo equalization and turbo decoding,” in *Proceeding of International Symposium on Turbo Codes*, Brest, France, September 1997, pp. 180-183.

[Rie98] S. Riedel, “MAP decoding of convolutional codes using reciprocal dual codes,” *IEEE Transactions on Information Theory,* vol. 44, no. 3, pp. 1176-1187, May 1998.

[RHV95] P. Robertson, P. Höher, and E. Villebrun, “A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain,” in *IEEE International Conference on Communications,* Seattle, WA, USA, June 1995, vol.2, pp. 1009-1013.

[RHV97] P. Robertson, P. Höher, and E. Villebrun, "Optimal and sub-optimal a posteriori algorithms suitable for turbo decoding," *European Transactions on Telecommunications*, vol.8, no.2, pp.119-25, August 1997.

[Rob94] P. Robertson, "Illuminating the structure of code and decoder for parallel concatenated recursive systematic (turbo) codes," in *IEEE Globecom*, San Francisco, USA, March 1994, vol. 3, pp. 1298-1303.

[Sha48] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pp. 623-656, October 1948.

[Sha49] C. E. Shannon, "Communication in the presence of noise," *Proc. of the Institute of Radio Engineers*, vol. 37, pp. 10-21, January 1949.

[Ste92] R. Steele, *Mobile Radio Communications*. Piscataway, NJ, USA: IEEE Press, 1992.

[SPT95] M. Siala, E. Papproth, K. H. Taieb, and T. K. Kaleh, "An iterative decoding scheme for serially concatenated convolutional codes," in *IEEE International Symposium on Information Theory*, Whistler, Canada, September 1995, p. 473.

[TaS87] H. Taub and D. L. Schilling, *Principles of Communication Systems*, 2nd edition. New York, USA: McGraw-Hill, 1987.

[Ung74] G. Ungerboeck, "Adaptive maximum-likelihood receiver for carrier-modulated data-transmission systems," *IEEE Transactions on Communications*, vol. 22, no. 5, pp. 624-636, May 1974.

[VaF98] P. Vary and T. Fingscheidt, "From soft decision channel decoding to soft-decision speech decoding," in *Proceedings of ITG-Fachtagung 'Codierung für Quelle, Kanal und Übertragung'*, Aachen, Germany, March 1998, pp. 155-162.

[Viz95] P. Vizmuller, *RF Design Guide - Systems, Circuits, and Equations*. Norwood, MA, USA: Artech House, Inc., 1995.

[Win98] J. H. Winters, "The diversity gain of transmit diversity in wireless systems with Rayleigh fading," *IEEE Transactions on Vehicular Technology*, vol. 47, no. 1, pp. 119-123, January 1998.

[WoJ65] J. M. Wozencraft and I. M. Jacobs, *Principles of Communication Engineering*. New York, USA: John Wiley & Sons, 1965.

[Wol78] J. K. Wolf, "Efficient maximum likelihood decoding of linear block codes using a trellis," *IEEE Transactions on Information Theory*, vol. 24, no. 1, pp. 76-80, January 1978.

[Wos98] P. Dueñas Wosta, "Quellengesteuerte Turbo-Detektion für Mobilfunksysteme," Diplomarbeit, Lehrstuhl für Nachrichtentechnik, TU München, 1998.