

Modulbezeichnung: Kanalcodierung (KaCo)
(Channel Coding)

5 ECTS

Modulverantwortliche/r: Clemens Stierstorfer
Lehrende: Clemens Stierstorfer

Startsemester: WS 2020/2021	Dauer: 1 Semester	Turnus: halbjährlich (WS+SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch oder Englisch

Lehrveranstaltungen:

Übungen zur Kanalcodierung (WS 2020/2021, Übung, 1 SWS, Clemens Stierstorfer)
Kanalcodierung (WS 2020/2021, Vorlesung, 3 SWS, Clemens Stierstorfer)

Empfohlene Voraussetzungen:

Es ist hilfreich, wenn die Studierenden die erlernten Algorithmen in eine Programmiersprache (C, Matlab usw.) umsetzen können.

It would be very helpful if the participants can implement the specified algorithms into a programming language (C, Matlab, etc.).

Es wird empfohlen, folgende Module zu absolvieren, bevor dieses Modul belegt wird:

Information Theory and Coding
Digitale Übertragung

Inhalt:

1 Introduction and Motivation 1.1 Definition, Related Fields 1.2 Basic Principles 1.2.1 Schemes 1.2.2 How to Add Redundancy 1.2.3 Applications 1.3 Historical Notes
2 Fundamentals of Block Coding 2.1 General Assumptions 2.2 Transmission Channels 2.2.1 Discrete-Time AWGN Channel 2.2.2 Binary Symmetric Channel (BSC) 2.2.3 Channels with Memory 2.3 Motivation for Coding 2.4 Fundamentals of Block Coding 2.4.1 Code and Encoding 2.4.2 Decoding
3 Introduction to Finite Fields I 3.1 Group 3.1.1 Orders of Elements and Cycles 3.1.2 Subgroups, Cosets 3.2 Field 3.3 Vector Spaces
4 Linear Block Codes 4.1 Generator Matrix 4.2 Distance Properties 4.3 Elementary Operations 4.4 Parity-Check Matrix 4.5 Dual Codes 4.6 Syndrome Decoding 4.7 Error Probability and Coding Gain 4.7.1 Error Detection 4.7.2 Error Correction - BMD 4.7.3 Error Correction - ML Decoding 4.7.4 Coding Gain 4.7.5 Asymptotic Results 4.8 Modifications of Codes 4.9 Bounds on the Minimum Distance 4.10 Examples for Linear Block Codes 4.10.1 Binary Hamming Codes ($q=2$) 4.10.2 Simplex Codes 4.10.3 Ternary Golay Code 4.10.4 Reed-Muller Codes
5 Linear Cyclic Codes 5.1 Modular Arithmetic 5.2 Generator Polynomial 5.3 Parity-Check Polynomial 5.4 Dual Codes 5.5 Discrete Systems over F_q 5.6 Encoders for Cyclic Codes 5.6.1 Generator Matrix 5.6.2 Non-Systematic Encoding 5.6.3 Systematic Encoding 5.6.4 Systematic Encoding Using $h(x)$ 5.7 Syndrome Decoding 5.7.1 Syndrome 5.7.2 Decoding Strategies 5.8 Examples for Linear Cyclic Block Codes 5.8.1 Repetition Code and Single Parity-Check Code 5.8.2 Binary Hamming Codes 5.8.3 Simplex Codes 5.8.4 Golay Codes 5.8.5 CRC Codes
6 Introduction to Finite Fields II 6.1 Extension Fields 6.2 Polynomials over Finite Fields 6.3 Primitive Element 6.4 Existence of Finite Fields 6.5 Finite Fields Arithmetic 6.6 Minimal Polynomials, Conjugate Elements, and Cyclotomic Cosets 6.7 Summary of Important Properties of Finite Fields 6.8 (Discrete) Fourier Transform over Finite Fields
7 BCH and RS Codes 7.1 The BCH Bound 7.2 Reed-Solomon Codes 7.3 BCH Codes 7.4 Algebraic Decoding of BCH Codes and RS Codes 7.4.1 Basic Idea 7.4.2 The Berlekamp-Massey Algorithm 7.5 Application: Channel Coding for CD and DVD 7.5.1 Error Correction for the CD 7.5.2 Error Correction for the DVD
8 Convolutional Codes 8.1 Discrete Systems over F 8.2 Trellis Coding 8.3 Encoders for Convolutional Codes 8.4 (Optimal) Decoding of Convolutional Codes 8.4.1 Maximum-Likelihood Sequence Estimation (MLSE) 8.4.2 Maximum A-Posteriori Symbol-by-Symbol Estimation
9 Codes with Iterative Decoding 9.1 State of the Art 9.2 Preliminaries 9.2.1 Check Equations 9.2.2 Repetition Code, Parallel Channels 9.2.3 Log-Likelihood Ratios (LLR) 9.3 Turbo Codes 9.4 LDPC Codes

Lernziele und Kompetenzen:

Das Modul Kanalcodierung umfasst eine Einführung in die Grundlagen der algebraischen, fehlerkorrigierenden Blockcodes sowie einen Einstieg in die Thematik der Faltungscodes. Iterativ decodierte Codeschemata wie Turbo-Codes und LDPC-Codes werden ebenfalls eingeführt. Im Einzelnen sind die Inhalte oben aufgeführt.

Die Studierenden definieren die Problematik der Kanalcodierung, grenzen sie von anderen Codierverfahren (z.B. der Quellencodierung) ab und kennzeichnen die unterschiedlichen Ansätze zur Fehlerkorrektur und -erkennung. Sie nennen Beispiele für Einsatzgebiete von Kanalcodierung und geben einen Überblick über die historische Entwicklung des Fachgebiets.

Die Studierenden erstellen Übertragungsszenarien für den Einsatz von Kanalcodierung bestehend aus Sender, Übertragungskanal und Empfänger und beachten dabei die Grundannahmen beim Einsatz von Blockcodes bzw. der Modellierung der Kanäle. Sie formulieren mathematische Beschreibungen der Encodierung sowie der optimalen Decodierung bzw. suboptimaler Varianten.

Die Studierenden beherrschen die Grundlagen fehlerkorrigierender linearer Blockcodes, beschreiben diese mathematisch korrekt mittels Vektoren und Matrizen über endlichen Körpern und implementieren und bewerten zugehörige Encoder- und Decoderstrukturen insbesondere Syndromdecoder. Dabei modifizieren sie Generatormatrizen, ermitteln Prüfmatrizen und erstellen Syndromtabellen. Sie schätzen die minimale Hammingdistanz von Codes mittels Schranken ab und können den erzielbaren Codegewinn erläutern. Sie kennen und benutzen beispielhaften Codefamilien (z.B. Hamming-Codes, Simplex-Codes, Reed-Muller-Codes).

Die Studierenden erkennen die Vorteile zyklischer linearer Blockcodes und beschreiben diese mit Polynomen über endlichen Körpern. Sie nutzen die Restklassenrechnung bzgl. Polynomen zur Umsetzung systematischer Encoder und zur Realisierung von Syndromdecodern mittels Schieberegisterschaltungen. Sie kennen beispielhafte Codefamilien.

Die Studierenden nutzen Primkörper, Erweiterungskörper, Minimalpolynome und Kreisteilungsklassen sowie die Spektraldarstellung über endlichen Körpern zur Realisierung von BCH- und Reed-Solomon-Codes gemäß der BCH-Schranke. Sie verstehen die Grundlagen der Decodierung von BCH- und Reed-Solomon-Codes. Sie skizzieren und erläutern die Kanalcodierkonzepte von CD und DVD.

Die Studierenden erklären die Unterschiede von Faltungscodes und Blockcodes, skizzieren anhand von tabellierten Generatorpolynomen zugehörige Encoder und erläutern diese. Sie erklären die Funktionsweise des optimalen Decoders (MLSE) und demonstrieren diese beispielhaft.

Die Studierenden verstehen die Grundlagen der iterativen Decodierung, insbesondere wenden sie die Grundlagen des Information Combining zur Kombination von verschiedenen Beobachtungen an. Sie verstehen die Bedeutung von Log-Likelihood-Ratios bei iterativen Decodieruvorgängen und berechnen diese. Sie skizzieren die grundlegenden Encoder- und Decoderstrukturen von Turbo-Codes und die Grundzüge der Codierung mit LDPC-Codes u.a. der Decodierung mittels Belief Propagation.

Die Vorlesung erfolgt wechselweise auf Deutsch oder Englisch (Winter/Sommer). Die zur Verfügung gestellten Unterlagen sind ausschließlich in Englisch gehalten. Die Studierenden verwenden entweder die englischen Fachtermini sicher oder kennen diese und drücken sich sicher mit den entsprechenden deutschen Fachbegriffen aus.

Die Umsetzung der angegebenen Algorithmen in eine Programmiersprache (C, Matlab usw.) sollten die Studierenden zu diesem Zeitpunkt des Studiums üblicherweise beherrschen. Übungen hierzu bleiben der Eigeninitiative überlassen.

Students define the problems of channel coding, how to distinguish it from other coding methods (such as source coding) and how to describe the various different approaches to error correction and detection. They are able to list example application areas of channel coding and give an overview of the historical development of the field. Furthermore, they describe and analyze transmission scenarios for the application of channel coding which consist of transmitter, transmission channel and receiver, taking into account the general assumptions for applying block codes or modeling the channels. They formulate mathematical descriptions of encoding, optimal decoding and sub-optimal methods.

Students illustrate the principles of error-correcting linear block codes and describe them mathematically using vectors and matrices over finite fields. They implement and analyze corresponding encoder and decoder structures, in particular syndrome decoders, and modify generator matrices, construct

test matrices and create syndrome tables. They estimate the minimum Hamming distance of codes using bounds and are able to explain the coding gain that can be achieved in individual cases. They analyze and use example code families (e.g. Hamming codes, simplex codes, Reed-Muller codes). Students explain the advantages of cyclic linear block codes and how to describe them with polynomials over finite fields. They apply polynomial modular arithmetic to implement systematic encoders and realize syndrome decoders using shift register circuits. They know and use exemplary code families. Students use prime fields, extension fields, minimal polynomials and cyclotomic cosets, and spectral representation over finite fields to implement BCH and Reed-Solomon codes using the BCH bound. They understand the foundations of decoding BCH and Reed-Solomon codes and how to sketch and explain the channel coding concepts of CDs and DVDs.

Students are able to describe the differences between convolutional codes and block codes, to sketch the respective encoders based on tabulated generator polynomials and to explain them. They are able to explain how optimal decoders (MLSE) work using examples.

Students sketch the foundations of iterative decoding. In particular, they apply methods of information combining to combine different observations. They use and calculate log-likelihood ratios in iterative decoding processes, sketch the basic encoding and decoding structures of turbo codes and the basics of coding using LDPC codes (including decoding using belief propagation).

Students either are able to use the English technical terms correctly or know them and are able to express themselves using the respective technical terms in German.

Literatur:

- J. Huber, R. Fischer, C. Stierstorfer: Folien zur Vorlesung
- M. Bossert: Kanalcodierung, Oldenbourg Wissenschaftsverlag, 3. Auflage, 2013
- M. Bossert: Channel Coding for Telecommunications, John Wiley & Sons, 1999
- B. Friedrichs: Kanalcodierung, Springer Verlag, 1996
- S.B. Wicker: Error Control Systems for Digital Communications and Storage, Prentice-Hall, 1995

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)

(Po-Vers. 2013 | TechFak | Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science) | Gesamtkonto | Wahlpflichtbereich Technisches Anwendungsfach | Information Technology - DT | Kanalcodierung)

Dieses Modul ist daneben auch in den Studienfächern "123#67#H", "Advanced Signal Processing & Communications Engineering (Master of Science)", "Berufspädagogik Technik (Bachelor of Science)", "Berufspädagogik Technik (Master of Education)", "Communications and Multimedia Engineering (Master of Science)", "Computational Engineering (Master of Science)", "Elektrotechnik, Elektronik und Informationstechnik (Bachelor of Science)", "Elektrotechnik, Elektronik und Informationstechnik (Master of Science)", "Information and Communication Technology (Master of Science)", "Informations- und Kommunikationstechnik (Bachelor of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Mathematik (Bachelor of Science)", "Medizintechnik (Master of Science)", "Wirtschaftsingenieurwesen (Master of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Kanalcodierung_ (Prüfungsnummer: 62701)

(englische Bezeichnung: Channel Coding)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

Erlaubte Hilfsmittel: nicht-programmierbarer Taschenrechner, ein Blatt Format A4 (oder entsprechende Fläche) mit eigenen, handschriftlichen Notizen; Prüfungssprache abhängig von der Wahl des Studierenden

Prüfungssprache: Deutsch oder Englisch

Erstablesung: WS 2020/2021, 1. Wdh.: SS 2021, 2. Wdh.: WS 2021/2022

1. Prüfer: Clemens Stierstorfer
