

# Lehrveranstaltung

## ACC - Advanced Channel Coding

---

Version: 2 | Letzte Änderung: 06.08.2019 21:55 | Entwurf: 0 | Status: vom verantwortlichen Dozent freigegeben

### ^ Allgemeine Informationen

<b>Langname</b>	Advanced Channel Coding
<b>Anerkennende LModule</b>	<a href="#">ACC MaCSN</a> , <a href="#">ACC MaTIN</a>
<b>Verantwortlich</b>	Prof. Dr. Uwe Dettmar Professor Fakultät IME
<b>Niveau</b>	Master
<b>Semester im Jahr</b>	Sommersemester
<b>Dauer</b>	Semester
<b>Stunden im Selbststudium</b>	78
<b>ECTS</b>	5
<b>Dozenten</b>	Prof. Dr. Uwe Dettmar Professor Fakultät IME
<b>Voraussetzungen</b>	- Grundwissen Lineare Algebra - Grundwissen Stochastik - Gute Programmierkenntnisse
<b>Unterrichtssprache</b>	englisch
<b>separate Abschlussprüfung</b>	Ja

## Abschlussprüfung

### Details

Prüfungsform: schriftliche Klausur (optional: Mündliche Prüfung)

- Dauer: 90 Minuten

- Arbeitsauftrag: i.d.R. 3 Aufgaben zu 10 Punkten (Bestehensgrenze: 12 Punkte) bestehend aus mehreren Teilaufgaben, die verschiedene Taxonomiestufen abprüfen.

- unterschiedliche Taxonomiestufen werden in der Punkteverteilung entsprechend ihrer

Komplexität und Schwierigkeit gewichtet

Besonderheit: Die Studierenden können 25% der Gesamtpunkte bereits aus den Zwischentests erwerben. Dies soll sie zu einer kontinuierlichen Mitarbeit motivieren.

### Mindeststandard

Grundwissen kann auf bekannte bzw. verwandte Probleme angewendet werden, Umsetzung teilweise fehlerhaft. (4,0)

### Prüfungstyp

Prüfungsform: schriftliche Klausur (optional: Mündliche Prüfung)

- Dauer: 90 Minuten
- Arbeitsauftrag: i.d.R. 3 Aufgaben zu 10 Punkten (Bestehensgrenze: 12 Punkte) bestehend aus mehreren Teilaufgaben, die verschiedene Taxonomiestufen abprüfen.
- unterschiedliche Taxonomiestufen werden in der Punkteverteilung entsprechend ihrer Komplexität und Schwierigkeit gewichtet

Besonderheit: Die Studierenden können 25% der Gesamtpunkte bereits aus den Zwischentests erwerben. Dies soll sie zu einer kontinuierlichen Mitarbeit motivieren.

## ^ Vorlesung / Übungen

### Lernziele

---

#### Kenntnisse

Vorlesung und Übungen werden in einer Lehrveranstaltung kombiniert. Nach der Vorstellung von neuem Lernstoff durch den Dozenten in Form von kurzen Blöcken wird dieser direkt von den Studierenden durch kurze Matlab- und Python-Übungen angewendet und vertieft. Längere Übungsaufgaben werden bereits zu Hause vorbereitet und die verschiedenen Lösungsvorschläge in der Präsenzveranstaltung besprochen.

Inhalte:

- Introduction
- Basic terms and definitions
- short history of channel coding
- System and channel models
- Review of binary error correcting block and convolutional Codes
- Generator and Parity check matrices,
- decoding principles, Trellis and Viterbi Algorithm
- Some principles on Information Theory
- Channel coding theorem
- Channel capacity and example calculations
- Cyclic Codes, Reed Solomon Codes
- Encoding and Decoding, Euklidean and Berlekamp-Massey -Algorithm for Decoding
- Basics on LDPC, Polar, and TURBO Codes
- iterative decoding, Sum Product Algorithm
- Recursive Convolutional Codes
- Performance comparison
- Basics on Space Time Coding
- Channel Model, Capacity improvement, Alamouti Scheme, STBC and STTC and their decoding

Die Studierenden lernen die o.g. Themen in der Vorlesung kennen, erwerben Grundwissen und vertiefen dieses durch Selbststudium mit Hilfe von Literatur, YouTube Videos und anderen Netzressourcen (selbstständige Informationsbeschaffung), sowie in Lerngruppen (Teamwork).

---

## Fertigkeiten

Durch kleine Übungsaufgaben und Programme wird in der Präsenzveranstaltung bereits ein aktiver Umgang mit den vorgestellten Verfahren ermöglicht. Umfangreichere Rechenaufgaben werden am Ende der Veranstaltung behandelt und die Lösungswege diskutiert, um dadurch den Studierenden relevante Problemstellungen vorzustellen und ihre Fähigkeit zur Lösungsfindung zu entwickeln.

Die Studierenden lernen darüber hinaus:

- nachrichtentechnische Systeme zu analysieren und deren Performanz zu ermitteln bzw. abzuschätzen.
- Verfahren der Quellen- und Kanalcodierung und Kryptologie zu vergleichen und zu bewerten
- Kenntnisse auf technische Problemstellungen anzuwenden

## Aufwand Präsenzlehre

Typ	Präsenzzeit (h/Wo.)
Vorlesung	2
Übungen (ganzer Kurs)	1
Übungen (geteilter Kurs)	0
Tutorium (freiwillig)	0

## Separate Prüfung

### Prüfungstyp

Übungsaufgabe mit fachlich / methodisch eingeschränktem Fokus unter Klausurbedingungen lösen

### Details

Der Erfolg des Lernprozesses sollte durch zwei vorlesungsbegleitende Zwischentests in Form von Multiple Choice und numerischen Aufgaben überprüft werden. Dies wird in elektronischer Form durch Verwendung der Ilias Lernplattform durchgeführt. Zur Steigerung der Motivation können die Studierenden aus diesen Tests bereits bis zu 10 Punkte sammeln, die in die Endnote eingehen.

### Mindeststandard

40% der Gesamtpunktzahl in jedem Test.

## ^ Praktikum

### Lernziele

---

### Fertigkeiten

Vorhandene Simulationsumgebungen wie z.B. die Matlab Communication Toolbox oder AFF3CT ([aff3ct.github.io](https://github.com/aff3ct/aff3ct)) werden verwendet um:

- theoretische Ergebnisse aus Vorlesung und Übung zu überprüfen
- FEC Algorithmen zu implementieren
- BER zu simulieren und die Performanz zu ermitteln, sowie Codes zu vergleichen
- Programme zum Bearbeiten verwandter Probleme anzupassen
- sich mit Standardprogrammen zur Simulation vertraut zu machen
- Teamwork zu üben

## Aufwand Präsenzlehre

Typ	Präsenzzeit (h/Wo.)
Praktikum	1
Tutorium (freiwillig)	0

## Separate Prüfung

### Prüfungstyp

Projektaufgabe im Team bearbeiten (z.B. im Praktikum)

### Details

Ergebnisse in einem 120 Minuten Vortrag präsentieren und ein kurzen Artikel schreiben, um die Fähigkeit zu stärken, in Zeitschriften oder auf Konferenzen zu veröffentlichen.

Das Paper wird von einer Partnergruppe reviewed.

### Mindeststandard

Die geforderten Simulationen wurden aufgesetzt, die Ergebnisse wurden präsentiert und das Paper wurde geschrieben. Präsentation und Artikel sind im Allgemeinen richtig und schlüssig.