



FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

Masterstudiengang

Medizintechnik

Modulhandbuch

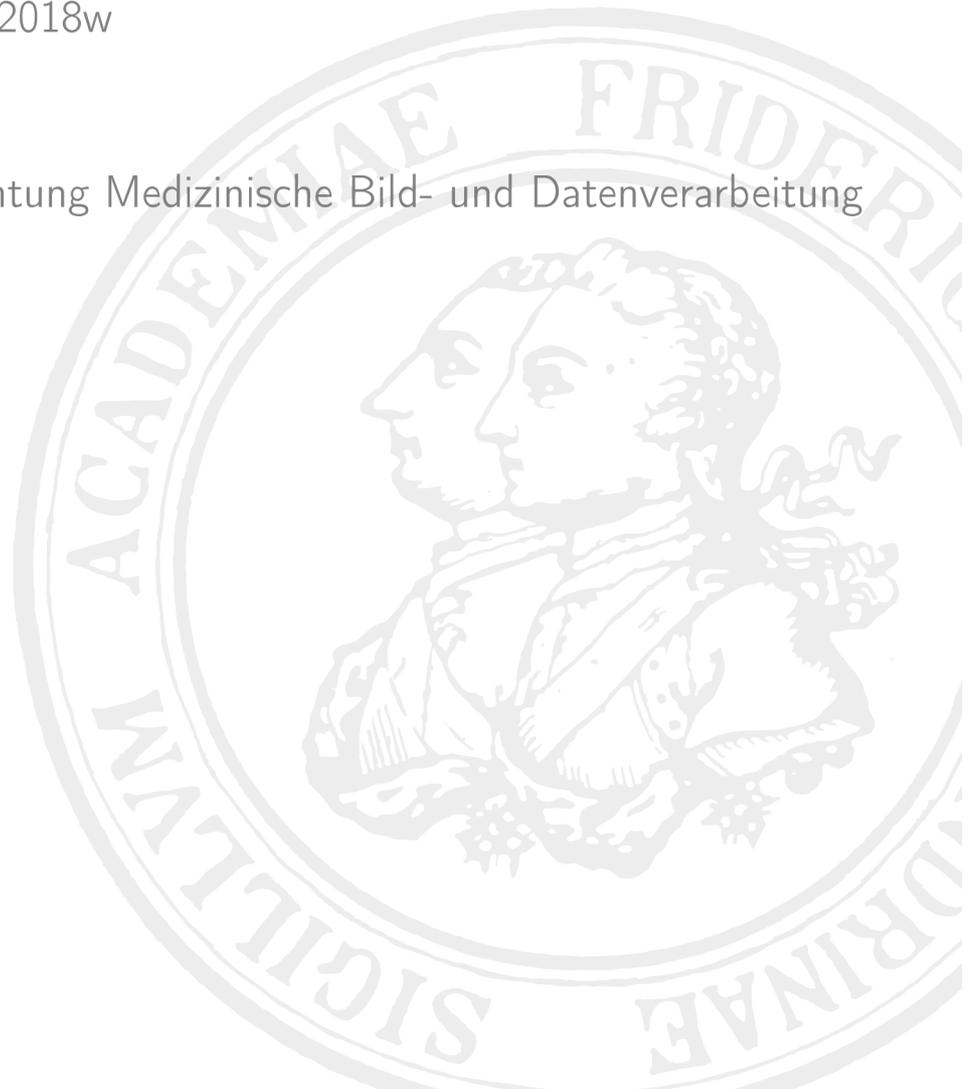
WS 2018/2019

Prüfungsordnungsversion: 2018w

Teilauszug Abschnitt

Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung

Modulhandbuch generiert aus *UnivIS*
Stand: 29.08.2021 20:37



Medizintechnik (Master of Science)

WS 2018/2019; Prüfungsordnungsversion: 2018w

M2 Ingenieurwissenschaftliche Kernmodule (BDV)

- Geometrische Modellierung - VU, 5 ECTS, Marc Stamminger, Roberto Grosso, WS 2018/2019 5
- 2018/2019
- Eingebettete Systeme (VU), 5 ECTS, Jürgen Teich, WS 2018/2019 7
- Numerik I für Ingenieure, 5 ECTS, Wilhelm Merz, J. Michael Fried, Nicolas Neuß, u.a., 9
- WS 2018/2019
- Digitale Signalverarbeitung, 5 ECTS, Walter Kellermann, Andreas Brendel, WS 2018/2019 10
- Functional Analysis for Engineers, 5 ECTS, Christoph Pflaum, WS 2018/2019 12
- Heterogene Rechnerarchitekturen Online, 5 ECTS, Marc Reichenbach, Thomas Heller, Johannes Hofmann, WS 2018/2019 13
- Maschinelles Lernen für Zeitreihen, 5 ECTS, Björn Eskofier, Oliver Amft, WS 2018/2019 15
- Reconfigurable Computing with Extended Exercises, 7.5 ECTS, Jürgen Teich, WS 2018/2019 17
- 2018/2019
- Information Theory and Coding, 5 ECTS, Ralf Müller, WS 2018/2019 20
- Künstliche Intelligenz I, 7.5 ECTS, Michael Kohlhase, WS 2018/2019 23
- Maschinelles Lernen für Zeitreihen Deluxe, 7.5 ECTS, Björn Eskofier, Oliver Amft, WS 2018/2019 25
- 2018/2019
- Konzeptionelle Modellierung, 5 ECTS, Richard Lenz, WS 2018/2019 27
- Computergraphik-VUP, 7.5 ECTS, Marc Stamminger, WS 2018/2019 29
- Deep Learning, 5 ECTS, Andreas Maier, Tobias Würfl, Vincent Christlein, Lennart Husvogt, WS 2018/2019 31
- Eingebettete Systeme mit erweiterter Übung, 7.5 ECTS, Jürgen Teich, Frank Hannig, WS 2018/2019 33
- 2018/2019
- Ereignisgesteuerte Systeme, 5 ECTS, Stefan Wildermann, WS 2018/2019 35
- Systemprogrammierung Vertiefung, 5 ECTS, Wolfgang Schröder-Preikschat, Jürgen Klei- 37
öder, WS 2018/2019
- Algorithms of Numerical Linear Algebra, 7.5 ECTS, Ulrich Rude, WS 2018/2019 39
- Reconfigurable Computing, 5 ECTS, Jürgen Teich, WS 2018/2019 41
- Echtzeitsysteme-V+Ü, 5 ECTS, Peter Ulbrich, WS 2018/2019 43
- Statistische Signalverarbeitung, 5 ECTS, Walter Kellermann, Alexander Schmidt, WS 2018/2019 47
- 2018/2019

- Digital Communications, 5 ECTS, Laura Cottatellucci, WS 2018/2019 50
- Kanalcodierung, 5 ECTS, Clemens Stierstorfer, WS 2018/2019 52
- Computergraphik-VU, 5 ECTS, Marc Stamminger, WS 2018/2019 55
- Cyber-Physical Systems, 5 ECTS, Torsten Klie, WS 2018/2019 58
- Domain-Specific and Resource-Aware Computing on Multicore Architectures, 5 ECTS, Frank Hannig, WS 2018/2019 60

- Pattern Recognition, 5 ECTS, Elmar Nöth, Sebastian Käßler, WS 2018/2019 62
- Pattern Recognition Deluxe, 7.5 ECTS, Elmar Nöth, Sebastian Käßler, WS 2018/2019 64

M3 Medizintechnische Kernmodule (BDV)

- Multidimensional Signals and Systems, 5 ECTS, Rudolf Rabenstein, WS 2018/2019 67
- Magnetic Resonance Imaging 1, 5 ECTS, Frederik Laun, Andreas Maier, Armin Nagel, WS 2018/2019 69

- Visual Computing in Medicine, 5 ECTS, Peter Hastreiter, Thomas Wittenberg, WS 2018/2019, 2 Sem. 70

- Biomedizinische Signalanalyse, 5 ECTS, Björn Eskofier, Heike Leutheuser, WS 2018/2019 73

- Diagnostic Medical Image Processing (VHB-Kurs), 5 ECTS, Andreas Maier, WS 2018/2019 75

M5 Medizintechnische Vertiefungsmodule (BDV)

UnivIS: 29.08.2021 20:37

3

- Test- und Analyseverfahren zur Softwareverifikation und -Validierung, 5 ECTS, Francesca Saglietti, WS 2018/2019 77

- Molecular Communications, 5 ECTS, Robert Schober, WS 2018/2019 79

- eBusiness Technologies und Evolutionäre Informationssysteme, 5 ECTS, Christoph P. Neumann, Richard Lenz, Florian Irmert, WS 2018/2019 80

- Computerunterstützte Messdatenerfassung, 5 ECTS, Reinhard Lerch, WS 2018/2019 82

- Lasers in Healthcare Engineering, 2.5 ECTS, Florian Klämpfl, WS 2018/2019 84

- Einführung in die IT-Sicherheit, 5 ECTS, Felix Freiling, WS 2018/2019 85

- Convex Optimization in Communications and Signal Processing, 5 ECTS, Wolfgang Gerstacker, WS 2018/2019 86

- Image, Video, and Multidimensional Signal Processing, 5 ECTS, André Kaup, WS 2018/2019 88

UnivIS: 29.08.2021 20:37

4

 Modulbezeichnung: Geometrische Modellierung - VU (GM-VU) 5 ECTS

(Geometric Modeling)

Modulverantwortliche/r: Marc Stamminger, Roberto Grosso

Lehrende: Marc Stamminger, Roberto Grosso

 Startsemester: WS 2018/2019 Dauer: 1 Semester Turnus: jährlich (WS)

Präsenzzeit: 60 Std. Eigenstudium: 90 Std. Sprache: Englisch

 Lehrveranstaltungen:

Geometric Modeling (WS 2018/2019, Vorlesung, 3 SWS, Roberto Grosso)

Tutorials to Geometric Modeling (WS 2018/2019, Übung, 1 SWS, Dominik Penk et al.)

Inhalt:

Die Vorlesung beschäftigt sich mit Methoden zur Modellierung dreidimensionaler Oberflächen. Typische Einsatzgebiete sind der rechnerunterstützte Entwurf (CAD, z.B. im Automobil- oder Flugzeugbau), die Rekonstruktion von Flächen aus Sensordaten oder die Konstruktion glatter Interpolationsflächen. Behandelt werden u.a. folgende Themen:

- Polynomkurven
- Bezierkurven, rationale Bezierkurven
- B-Splines
- Tensorproduktflächen
- Bezier-Dreiecksflächen
- polygonale Flächen
- Subdivision-Verfahren

This lecture is concerned with different aspects of modelling three-dimensional curves and surfaces. Typical areas of application are computer-aided design (CAD), reconstruction of surfaces from sensor data (reverse engineering) and construction of smooth interpolants. The lecture covers the following topics:

- polynomial curves
- Bézier curves, rational Bézier curves
- B-splines
- tensor product surfaces
- triangular Bézier surfaces
- polyhedral surfaces

Lernziele und Kompetenzen: Die Studierenden

- erklären die Begriffe Polynomial-, Bezierkurven und B-Splines
- klassifizieren und veranschaulichen die unterschiedlichen Auswertung- und Subdivision-Verfahren für Bezier-Kurven und B-Splines
- veranschaulichen und ermitteln die Eigenschaften von Bezierkurven, rationalen Bezierkurven und B-Splines
- beschreiben Tensorproduktflächen und skizzieren Auswertungsalgorithmen
- erklären polygonale Flächen und Subdivision-Verfahren und veranschaulichen ihre Unterschiede und Eigenschaften
- lernen gängige Datenstrukturen zur Darstellung polygonaler Flächen kennen
- wenden die Verfahren der Geometrischen Modellierung an unterschiedlichen Beispiele an
- berechnen Bezierkurven und B-Splines
- führen Subdivision-Verfahren aus

Educational objectives and skills:

Students should be able to

- explain the meaning of the terms Polynomial and Bezier curves and B-Splines
 - classify and illustrate the different evaluation and subdivision methods for Bezier curves and BSplines
 - describe and establish the properties of Bezier curves, rational Bezier curves and B-Splines
 - describe tensor product surfaces and illustrated evaluation algorithms
 - explain polygonal surfaces and subdivision algorithms and depict their properties and differences
 - get used with common data structures to represent polygonal surfaces
 - apply geometric modeling algorithms to representative examples
 - compute Bezier curves and B-Splines
 - implement subdivision algorithms Literatur:
 - Hoschek, Lasser: Grundlagen der Geometrischen Datenverarbeitung
 - Farin: Kurven und Flächen im Computer Aided Geometric Design
 - de Boor: A Practical Guide to Splines
 - Bartels, Beatty, Barsky: Splines for Use in Computer Graphics and Geometric Modeling
 - Abramowski, Müller: Geometrisches Modellieren
-

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

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

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Bachelor of Arts (2 Fächer))", "Informatik (Bachelor of Science)", "Informatik (Master of Science)", "Mathematik (Bachelor of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Geometric Modeling (Vorlesung mit Übung) (Prüfungsnummer: 796399)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 0%

weitere Erläuterungen:

Klausur in elektronischer Form mit einem Anteil im Antwort-Wahl-Verfahren

Erstablingung: WS 2018/2019, 1. Wdh.: SS 2019

1. Prüfer: Roberto Grosso, 2. Prüfer: Marc Stamminger

Modulbezeichnung:	Eingebettete Systeme (VU) (ES-VU) (Embedded Systems (VU))	5 ECTS
Modulverantwortliche/r:	Jürgen Teich	
Lehrende:	Jürgen Teich	
Startsemester: WS 2018/2019	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

Lehrveranstaltungen:

Eingebettete Systeme (WS 2018/2019, Vorlesung, 2 SWS, Jürgen Teich et al.)

Übung zu Eingebettete Systeme (WS 2018/2019, Übung, 2 SWS, Jörg Fickenscher et al.)

Inhalt:

Schwerpunkt der Vorlesung ist der Entwurf und die Implementierung eingebetteter Systeme unter Einsatz formaler Methoden und rechnergestützter Entwurfsverfahren.

Unter eingebetteten Systemen versteht man Rechensysteme, die auf einen Anwendungsbereich zugeschnitten (z.B. mobile Kommunikationsgeräte, Chipkartensysteme, Industriesteuerungen, Unterhaltungselektronik, Medizintechnik) und in einen technischen Kontext eingebunden sind. Das

große Interesse am systematischen Entwurf von heterogenen eingebetteten Systemen ist verursacht durch die steigende Vielfalt und Komplexität von Anwendungen für eingebettete Systeme, die Notwendigkeit, Entwurfs- und Testkosten zu senken sowie durch Fortschritte in Schlüsseltechnologien (Mikroelektronik, formale Methoden).

Content:

The focus of the lecture is the design and implementation of embedded systems using formal methods and computer-aided design techniques.

Embedded systems are computing systems tailored for a particular application (e.g., mobile communication devices, smart card systems, industrial control, consumer electronics, medical technology) and integrated into a technical context. The keen interest in the systematic design of heterogeneous embedded systems is driven by the increasing diversity and complexity of embedded system applications, the need to reduce design and test costs, and advances in key technologies (microelectronics, formal methods).

Lernziele und Kompetenzen:

Fachkompetenz

Wissen

- Die Studierenden setzen sich mit einem aktuellen Forschungsgebiet auseinander. / *The students deal with a current field of research.*

Verstehen

- Die Studierenden verstehen grundlegende Konzepte des Entwurfs eingebetteter Systeme. / *The students become familiar with the fundamental concepts of designing of embedded systems.*

Anwenden

- Die Studierenden wenden grundlegende Algorithmen an zur Analyse und Optimierung von Hardware-Architekturen und Echtzeit-Softwaresystemen. / *The students apply basic algorithms to analyze and optimize hardware architectures and real-time software systems.*
- Die Studierenden erfassen den Hardware/Software-Entwurf von Systemen mit harten Beschränkungen. / *The students understand the hardware/software design of hard-constrained systems.*

Literatur:

- Buch zur Vorlesung
 - Vorlesungsskript (Zugriff nur innerhalb des Uni-Netzwerks möglich)
-

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

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

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern " Communications Engineering (Master of Science)", "123#67#H", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Informatik (Bachelor of Arts (2 Fächer))", "Informatik (Bachelor of Science)", "Informatik (Master of Science)", "Informations- und Kommunikationstechnik (Bachelor of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Mathematik (Bachelor of Science)", "Mechatronik (Bachelor of Science)", "Medizintechnik (Bachelor of Science)", "Wirtschaftsinformatik (Bachelor of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Schriftliche Prüfung Eingebettete Systeme (Prüfungsnummer: 44101)

(englische Bezeichnung: Written Examination Embedded Systems)

Prüfungsleistung, Klausur

Anteil an der Berechnung der Modulnote: 100%
weitere Erläuterungen:
Die Modulnote ergibt sich aus der Klausurnote.

Erstablingung: WS 2018/2019, 1. Wdh.: SS 2019
1. Prüfer: Jürgen Teich

Organisatorisches:

Die Auswahl dieses Moduls schließt die Auswahl des Moduls „Eingebettete Systeme mit erweiterter Übung“ aus.

Erstablingung: WS 2018/2019, 1. Wdh.: SS 2019 (nur für Wiederholer) 1.
 Prüfer: J. Michael Fried

Modulbezeichnung:	Digitale Signalverarbeitung (DSV) (Digital Signal Processing)	5 ECTS
Modulverantwortliche/r:	Walter Kellermann	
Lehrende:	Walter Kellermann, Andreas Brendel	
Startsemester: WS 2018/2019	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Englisch

Lehrveranstaltungen:

Digitale Signalverarbeitung (WS 2018/2019, Vorlesung, 3 SWS, Walter Kellermann)
 Ergänzungen und Übungen zur Digitalen Signalverarbeitung (WS 2018/2019, Übung, 1 SWS, Andreas Brendel et al.)
 Tutorium zur Digitalen Signalverarbeitung (WS 2018/2019, optional, Tutorium, 1 SWS, Andreas Brendel)

Empfohlene Voraussetzungen:

Vorlesung Signale und Systeme I & II

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

Signale und Systeme II
 Signale und Systeme I

Inhalt:

The course assumes familiarity with basic theory of discrete-time deterministic signals and linear systems and extends this by a discussion of the properties of idealized and causal, realizable systems (e.g., lowpass, Hilbert transformer) and corresponding representations in the time domain, frequency domain, and z-domain. Thereupon, design methods for recursive and nonrecursive digital filters are discussed. Recursive systems with prescribed frequency-domain properties are obtained by using design methods for Butterworth filters, Chebyshev filters, and elliptic filters borrowed from analog filter design. Impulse-invariant transform and the Prony-method are representatives of the considered designs with prescribed time-domain behaviour. For nonrecursive systems, we consider the Fourier approximation in its original and its modified form introducing a broad selection of windowing functions. Moreover, the equiripple approximation is introduced based on the Remez-exchange algorithm.

Another section is dedicated to the Discrete Fourier Transform (DFT) and the algorithms for its fast realizations ('Fast Fourier Transform'). As related transforms we introduce cosine and sine transforms. This is followed by a section on nonparametric spectrum estimation. Multirate systems and their efficient realization as polyphase structures form the basis for describing analysis/synthesis filter banks and discussing their applications.

The last section is dedicated to investigating effects of finite wordlength as they are unavoidable in any realization of digital signal processing systems.

A corresponding lab course on DSP will be offered in the winter term.

Lernziele und Kompetenzen:

Die Studierenden

- analysieren zeitdiskrete lineare zeitinvariante Systeme durch Ermittlung der beschreibenden Funktionen und Parameter
- wenden grundlegende Verfahren zum Entwurf zeitdiskreter Systeme an und evaluieren deren Leistungsfähigkeit

- verstehen die Unterschiede verschiedener Methoden zur Spektralanalyse und können damit vorgegebene Signale analysieren
- verstehen die Beschreibungsmethoden von Multiraten-Systemen und wenden diese zur Beschreibung von Filterbänken an
- kennen elementare Methoden zur Analyse von Effekten endlicher Wortlängen und wenden diese auf zeitdiskrete lineare zeitinvariante Systeme an

The students

- analyze discrete-time linear time-invariant systems by determining the describing function and parameters
- apply fundamental approaches for the design of discrete-time systems and evaluate their performance
- understand the differences between various methods for spectral analysis and apply them to the analysis of given signals
- understand methods to represent multirate systems and apply them for the representation of filter banks
- know basic methods for the analysis of finite word length effects and apply them to discrete-time linear time-invariant systems.

Literatur:

Empfohlene Literatur/ Recommended Reading:

1. J.G. Proakis, D.G. Manolakis: Digital Signal Processing. 4th edition. Prentice Hall, Englewood Cliffs, NJ, 2007.
 2. A.V. Oppenheim, R.V. Schaffer: Digital Signal Processing. Prentice Hall, Englewood Cliffs, NJ, 1975.
 3. K.D. Kammeyer, K. Kroschel: Digitale Signalverarbeitung: Filterung und Spektralanalyse mit MATLAB®-Übungen . 8. Aufl. Teubner, Stuttgart, 2012
-

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

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

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "123#67#H", "Berufspädagogik Technik (Bachelor of Science)", "Berufspädagogik Technik (Master of Education)", "Communications and Multimedia Engineering (Master of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Elektrotechnik, Elektronik und Informationstechnik (Bachelor of Science)", "Elektrotechnik, Elektronik und Informationstechnik (Master of Science)", "Informatik (Master of Science)", "Informations- und Kommunikationstechnik (Bachelor of Science)", "Mathematik (Bachelor of Science)", "Medizintechnik (Bachelor of Science)", "Technomathematik (Bachelor of Science)", "Wirtschaftsingenieurwesen (Bachelor of Science)", "Wirtschaftsingenieurwesen (Master of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Digitale Signalverarbeitung (Prüfungsnummer: 35001)

(englische Bezeichnung: Digital Signal Processing)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90 Anteil
an der Berechnung der Modulnote: 100%

Erstablingung: WS 2018/2019, 1. Wdh.: SS 2019

1. Prüfer: Walter Kellermann

Modulbezeichnung: Functional Analysis for Engineers (FuncAnEng)
(Functional Analysis for Engineers)

5 ECTS

 Modulverantwortliche/r: Christoph Pflaum

 Lehrende: Christoph Pflaum

Startsemester: WS 2018/2019 Dauer: 1 Semester Turnus: jährlich (WS)

 Präsenzzeit: 60 Std. Eigenstudium: 90 Std. Sprache: Englisch

Lehrveranstaltungen:

 Functional Analysis for Engineers (WS 2018/2019, Vorlesung mit Übung, 2 SWS, Christoph Pflaum)
 Recitation of Functional Analysis for Engineers (WS 2018/2019, Übung, 2 SWS, Phillip Rall)

Inhalt:

- vector spaces, norms, principal axis theorem
- Banach spaces, Hilbert spaces
- Sobolev spaces
- theory of elliptic differential equations
- Fourier transformation
- distributions

Lernziele und Kompetenzen:

Students learn advanced methods in linear algebra and basic concepts of functional analysis. Furthermore, students learn applications in solving partial differential equations. The course teaches abstract mathematical structures. Literatur:

- Lehrbuch: Dobrowolski, Angewandte Funktionalanalysis, Springer 2006.
-

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

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

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "123#67#H", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Informatik (Master of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Functional Analysis for Engineers (Prüfungsnummer: 575129)

(englische Bezeichnung: Functional Analysis for Engineers)

Prüfungsleistung, Klausur, Dauer (in Minuten): 60 Anteil
 an der Berechnung der Modulnote: 100% weitere

Erläuterungen:

Die Note ergibt sich aus einer 60minütigen Klausur. Voraussetzung zur Teilnahme an der Klausur ist die erfolgreiche Bearbeitung der Übungsaufgaben.

Erstablegung: WS 2018/2019, 1. Wdh.: SS 2019

 1. Prüfer: Wilhelm Merz

Modulbezeichnung:	Heterogene Rechnerarchitekturen Online (HETRON) (Heterogeneous Computing Architectures Online)	5 ECTS
--------------------------	---	--------

Modulverantwortliche/r: Dietmar Fey, Marc Reichenbach

 Lehrende: Thomas Heller, Marc Reichenbach, Johannes Hofmann

Startsemester: WS 2018/2019 Dauer: 1 Semester Turnus: halbjährlich (WS+SS)

 Präsenzzeit: k.A. Std. Eigenstudium: 150 Std. Sprache: Englisch

Inhalt:

Whereas heterogeneous architectures and parallel computing has filled an academic niche in the past it has become now a commodity technique with the rising of multi-core processors and programmable graphic cards. Even FPGAs play a role hereby in a certain extent due to their increasing importance as accelerator hardware what is clearly observable in the scientific community. However, on one side parallel hardware like multi-core and GPUs are now available nearly for everybody and not only for a selected selection of people, who have access to a parallel supercomputer. On the other side the knowledge about programming of this commodity hardware, and we mean here in particular hardwareorientated programming in order to squeeze out all offered GFlops and TFlops of such hardware, is still missing as well as the knowledge about the architecture details. To overcome this lack we offer this course HETRON.

The e-learning course HETRON for the exploitation of parallel and heterogeneous computer architectures) focuses on two main topics which are closely related to each other. This concerns on one side the benefits of using different kinds of multi-core processors and parallel architectures built-up on base of these multicore processors. These architectures differ among each other in the number and in the complexity of its single processing nodes. We distinguish between systems consisting of a large number of simpler, so called fine-grained, processor cores vs. systems consisting of a smaller number of more complex, so called coarse-grained, processor cores. On the other side we lay our focus on that we want to do with these different heterogeneous parallel architectures, namely the execution of parallel programs. Of course this requires the use of parallel programming languages and environments, like CUDA or OpenMP. However, besides these questions of using the right syntax and the right compiler switches to optimize a parallel program it is a pre-requisite to understand how parallel computing really works. This refers (i) to the comprehension which basic mechanisms of parallel computing exist, (ii) where are the limits of getting more performance with parallel computing and (iii) in what context stand these mechanisms to heterogeneous architectures. In other words it handles the question which architecture is the best one for a certain parallelization technique. To teach these three topics, is one main goal we pursuit with the course HETRON, and of course, this more fundamental basics of heterogeneous and parallel computing have to be proven by means of concrete application examples to deepen the acquired knowledge about heterogeneous architectures and parallel computing principles.

Lernziele und Kompetenzen:

Die Studierenden ...

...verstehen die Notwendigkeit sowie grundlegende Anwendungsfälle für heterogene Rechnerarchitekturen.

...können den grundlegenden Aufbau und das Zusammenspiel der Komponenten heterogener Rechnerarchitekturen erklären. ...erläutern grundsätzliche Parallelisierungsprinzipien wie Amdahls Law, HighPerformance- und High-Throughput-Computing sowie Parallelisierungsstrategien. ...können einfache Programme mit Hilfe der vermittelten Parallelisierungsprinzipien (Amdahls Law, High-Performance- und High-Throughput-Computing) analysieren und entsprechende Parallelisierungsstrategien entwickeln. ...erklären den Aufbau sowie Stärken und Schwächen von verschiedenen Architekturen wie CPUs, GPUs, Many-Core Prozessoren und FPGAs.

...implementieren ausgewählte Anwendungsbeispiele (SHA256 Algorithmus, Ising-Modell und FastFourier-Transformation) auf oben genannte Architekturen.

...erforschen und bewerten verschiedener Parallelisierungstechniken in Abhängigkeit der Anwendung und der Architektur.

...erläutern die Grundlagen des Grid- und Cloud-Computings

 ...sind in der Lage parallele Berechnungen (SHA256) im Grid umzusetzen.

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

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

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Informatik (Bachelor of Arts (2 Fächer))", "Informatik (Bachelor of Science)", "Informatik (Master of Science)", "Mathematik (Bachelor of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Heterogene Rechnerarchitekturen Online (Prüfungsnummer: 275245)

(englische Bezeichnung: Heterogeneous Computing Architectures Online)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 0% Prüfungssprache: Deutsch oder Englisch

Erstablingung: WS 2018/2019, 1. Wdh.: SS 2019

1. Prüfer: Dietmar Fey

Modulbezeichnung:	Maschinelles Lernen für Zeitreihen (MLTS) (Machine Learning for Time Series)	5 ECTS
Modulverantwortliche/r:	Björn Eskofier, Oliver Amft	
Lehrende:	Björn Eskofier, Oliver Amft	
Startsemester: WS 2018/2019	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Englisch

Lehrveranstaltungen:

Maschinelles Lernen für Zeitreihen (WS 2018/2019, Vorlesung, 2 SWS, Björn Eskofier et al.)
 Maschinelles Lernen für Zeitreihen Übung (WS 2018/2019, Übung, 2 SWS, Nooshin Haji Ghassemi et al.)

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

Pattern Analysis
 Introduction to Pattern Recognition
 Pattern Recognition

Inhalt:

Die Vorlesung vermittelt Konzepte des Maschinellen Lernens speziell im Hinblick auf Anwendungen bei Zeitreihen. Es handelt sich hier um eine Spezialisierungsvorlesung, eine erfolgreiche Absolvierung der Vorlesungen „IntroPR“ und/oder „Pattern Recognition“/„Pattern Analysis“ wird empfohlen. Konzepte, die in „IntroPR“ vermittelt werden, werden hier als Grundwissen vorausgesetzt. Die folgenden Themen werden in der Vorlesung behandelt:

- Ein Überblick über die Anwendungsgebiete der Zeitreihenanalyse
- Methodische Grundlagen des Maschinellen Lernens (ML) für die Analyse von Zeitreihen, beispielsweise Gauß-Prozesse, Monte-Carlo Sampling und Deep Learning
- Design, Implementierung und Evaluation von ML Methoden, um Probleme in Zeitreihen zu adressieren
- Arbeitstechniken in bekannten Toolboxes zur Implementierung von relevanten Methoden, beispielsweise Tensorflow/Keras

Content

Aim of the lecture is to teach Machine learning (ML) methods for a variety of time series applications. The following topics will be covered:

- An overview of applications of time series analysis
- Fundamentals of Machine learning (ML) methods, such as Gaussian processes, Monte Carlo sampling methods and deep learning, for time series analysis
- Design, implementation and evaluation of ML methods in order to address time series problems
- Working with widely-used toolboxes that can be used for implementation of ML methods, such as Tensorflow or Keras

Lernziele und Kompetenzen:

- Die Studierenden sollen ein Verständnis für Zeitreihenprobleme und deren Lösungen in Applikationsgebieten der Industrie, Medizin, dem Finanzwesen, etc. entwickeln
- Die Studierenden erlernen Konzepte des Maschinellen Lernens im Allgemeinen und deren Anwendung auf Zeitreihen im Besonderen
- Die Studierenden erlernen die Charakteristika von Zeitreihendaten und werden zur Entwicklung und Implementierung von ML-Methoden angeleitet, um solche Daten in konkreten Fragestellungen zu modellieren, manipulieren und vorherzusagen.

Learning Objectives

- Students develop an understanding of concepts of time series problems and their wide applications in industry, medicine, finance, etc.
 - Students learn concepts of machine learning (ML) methods in general and tackling time series problems in particular
 - Students understand the characteristics of time series data and will be capable of developing and implementing ML methods to model, predict and manipulate such data in concrete problems
Studon: https://www.studon.fau.de/studon/goto.php?target=crs_2202195 Literatur:
 - Machine Learning: A Probabilistic Perspective, Kevin Murphy, MIT press, 2012
 - The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Trevor Hastie, Robert Tibshirani, Jerome Friedman, Springer, 2009
 - Deep Learning, Ian Goodfellow and Yoshua Bengio and Aaron Courville, MIT Press, 2016
 - Reinforcement Learning: An Introduction, Richard S. Sutton and Andrew G. Barto, MIT press, 1998
-

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

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

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Elektrotechnik, Elektronik und Informationstechnik (Bachelor of Science)", "Elektrotechnik, Elektronik und Informationstechnik (Master of Science)", "Informatik (Bachelor of Arts (2 Fächer))", "Informatik (Bachelor of Science)", "Informatik (Master of Science)", "Mathematik (Bachelor of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Maschinelles Lernen für Zeitreihen (Prüfungsnummer: 428256)

(englische Bezeichnung: Machine Learning for Time Series)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100% Prüfungssprache: Deutsch oder Englisch

Erstablingung: WS 2018/2019, 1. Wdh.: SS 2019

1. Prüfer: Björn Eskofier

Modulbezeichnung: Reconfigurable Computing with Extended Exercises (RC-VEU) 7.5 ECTS

(Reconfigurable Computing with Extended Exercises)

Modulverantwortliche/r: Jürgen Teich

Lehrende: Jürgen Teich

Startsemester: WS 2018/2019 Dauer: 1 Semester Turnus: jährlich (WS)

Präsenzzeit: 90 Std. Eigenstudium: 135 Std. Sprache: Englisch

Lehrveranstaltungen:

Reconfigurable Computing (WS 2018/2019, Vorlesung, 2 SWS, Jürgen Teich)

Exercises to Reconfigurable Computing (WS 2018/2019, Übung, 2 SWS, Faramarz Khosravi et al.)

Extended Exercises to Reconfigurable Computing (WS 2018/2019, Übung, Martin Letras)

Inhalt:

Content:

Reconfigurable (adaptive) computing is a novel yet important research field investigating the capability of hardware to adapt to changing computational requirements such as emerging standards,

late design changes, and even to changing processing requirements arising at run-time. Reconfigurable computing thus benefits from a) the programmability of software similar to the Von Neumann computer and b) the speed and efficiency of parallel hardware execution.

The purpose of the course reconfigurable computing is to instruct students about the possibilities and rapidly growing interest in adaptive hardware and corresponding design techniques by providing them the necessary knowledge for understanding and designing reconfigurable hardware systems and studying applications benefiting from dynamic hardware reconfiguration.

After a general introduction about benefits and application ranges of reconfigurable (adaptive) computing in contrast to general-purpose and application-specific computing, the following topics will be covered:

- Reconfigurable computing systems: Introduction of available technology including fine grained look up table (LUT-) based reconfigurable systems such as field programmable gate arrays (FPGA) as well as newest coarse grained architectures and technology.
- Design and implementation: Algorithms and steps (design entry, functional simulation, logic synthesis, technology mapping, place and route, bit stream generation) to implement (map) algorithms to FPGAs. The main focus lies on logic synthesis algorithms for FPGAs, in particular LUT technology mapping.
- Temporal partitioning: techniques to reconfigure systems over time. Covered are the problems of mapping large circuits which do not fit one single device. Several temporal partitioning techniques are studied and compared.
- Temporal placement: Techniques and algorithms to exploit the possibility of partial and dynamic (run-time) hardware reconfiguration. Here, OS-like services are needed that optimize the allocation and scheduling of modules at run-time.
- On-line communication: Modules dynamically placed at run-time on a given device need to communicate as well as transport data off-chip. State-of-the-art techniques are introduced how modules can communicate data at run-time including bus-oriented as well as network-on-a-chip (NoC) approaches.
- Designing reconfigurable applications on Xilinx Virtex FPGAs: In this part, the generation of partial bitstreams for components to be placed at run-time on Xilinx FPGAs is introduced and discussed including newest available tool flows.
- Applications: This section presents applications benefiting from dynamic hardware reconfiguration. It covers the use of reconfigurable systems including rapid prototyping, reconfigurable supercomputers, reconfigurable massively parallel computers and studies important application domains such as distributed arithmetic, signal processing, network packet processing, control design, and cryptography.

Lernziele und Kompetenzen:

Learning objectives and competencies:

Fachkompetenz

Wissen

- The students know to exploit run-time reconfigurable design methodologies for adaptive applications.

Verstehen

- The students understand the mapping steps, and optimization algorithms.
- The students classify different types and kinds of reconfigurable hardware technologies available today.
- The students clarify pros and cons of reconfigurable computing technology.
- The students summarize applications benefiting from reconfigurable computing.

Anwenden

- The students apply design tools for implementation of circuits and systems-on-a-chip (SoC) on FPGAs during practical training.

Sozialkompetenz

- The students perform group work in small teams during practical training.

Literatur:

- The Hamburg VHDL Archive (see Documentation link for free books)
 - Interactive VHDL Tutorial with 150 examples from ALDEC
 - Easy FPGA tutorials, projects and boards
 - Xilinx WebPack ISE and Modelsim MXE (free FPGA synthesis tool and free VHDL simulator)
 - Symphone EDA free VHDL simulator (select FREE Edition license)
 - Icarus open-source Verilog simulator
-

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

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

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Bachelor of Arts (2 Fächer))", "Informatik (Bachelor of Science)", "Informatik (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Mathematik (Bachelor of Science)", "Mechatronik (Bachelor of Science)", "Mechatronik (Master of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Reconfigurable Computing (Lecture with Extended Exercises) (Prüfungsnummer: 714289)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100% weitere

Erläuterungen:

Oral examination (Duration: 30 min) + Successful attendance of the extended exercises (mandatory) + Successful completion of all tasks in the exercises (mandatory) Final grade of the module is determined by the oral examination.

Erstablingung: WS 2018/2019, 1. Wdh.: SS 2019

1. Prüfer: Jürgen Teich

Organisatorisches:

Selection of this module prohibits the selection of the module "Reconfigurable Computing (RC-VU)" by the student. Bemerkungen:

Reconfigurable computing is an interdisciplinary field of research between computer science and electrical engineering on a 4 SWS (4 hours/week) basis. Lecture and Exercises will give 5 ECTS, the FPGA & VHDL labs 2.5 ECTS.

Modulbezeichnung:	Information Theory and Coding (ITC) (Information Theory and Coding)	5 ECTS
Modulverantwortliche/r:	Ralf Müller	
Lehrende:	Ralf Müller	
Startsemester: WS 2018/2019	Dauer: 1 Semester	Turnus: halbjährlich (WS+SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Englisch
Lehrveranstaltungen:		
Information Theory and Coding (WS 2018/2019, Vorlesung, 3 SWS, Ralf Müller)		
Tutorial for Information Theory and Coding (WS 2018/2019, Übung, 1 SWS, Ali Beryhi)		

Inhalt:

1. Introduction: binomial distribution, (7,4)-Hamming code, parity-check matrix, generator matrix 2. Probability, entropy, and inference: entropy, conditional probability, Bayes' law, likelihood, Jensen's inequality
3. Inference: inverse probability, statistical inference
4. The source coding theorem: information content, typical sequences, Chebychev inequality, law of large numbers
5. Symbol codes: unique decidability, expected codeword length, prefix-free codes, Kraft inequality, Huffman coding
6. Stream codes: arithmetic coding, Lempel-Ziv coding, Burrows-Wheeler transform
7. Dependent random variables: mutual information, data processing lemma
8. Communication over a noisy channel: discrete memory-less channel, channel coding theorem, channel capacity
9. The noisy-channel coding theorem: jointly-typical sequences, proof of the channel coding theorem, proof of converse, symmetric channels
10. Error-correcting codes and real channels: AWGN channel, multivariate Gaussian pdf, capacity of AWGN channel
11. Binary codes: minimum distance, perfect codes, why perfect codes are bad, why distance isn't everything
12. Message passing: distributed counting, path counting, low-cost path, min-sum (=Viterbi) algorithm
13. Exact marginalization in graphs: factor graphs, sum-product algorithm
14. Low-density parity-check codes: density evolution, check node degree, regular vs. irregular codes, girth
15. Lossy source coding: transform coding and JPEG compression
-
1. Einleitung: Binomialverteilung, (7,4)-Hamming-Code, Paritätsmatrix, Generatormatrix
2. Wahrscheinlichkeit, Entropie und Inferenz: Entropie, bedingte Wahrscheinlichkeit, Bayes'sches Gesetz, Likelihood, Jensen'sche Ungleichung
3. Inferenz: Inverse Wahrscheinlichkeit, statistische Inferenz
4. Das Quellencodierungstheorem: Informationsgehalt, typische Folgen, Tschebyschev'sche Ungleichung, Gesetz der großen Zahlen
5. Symbolcodes: eindeutige Dekodierbarkeit, mittlere Codewortlänge, präfixfreie Codes, Kraft'sche Ungleichung, Huffmancodierung
6. Stromcodes: arithmetische Codierung, Lempel-Ziv-Codierung, Burrows-Wheeler-Transformation
7. Abhängige Zufallsvariablen: Transinformation, Datenverarbeitungslemma
8. Kommunikation over gestörte Kanäle: diskreter gedächtnisloser Kanal, Kanalcodierungstheorem, Kanalkapazität
9. Das Kanalcodierungstheorem: verbundtypische Folgen, Beweis des Kanalcodierungstheorems, Beweis des Umkehrsatzes, symmetrische Kanäle

10. Fehlerkorrigierende Codes und reale Kanäle: AWGN-Kanal, mehrdimensionale Gauß'sche WDF, Kapazität des AWGN-Kanals
11. Binäre Codes: Minimaldistanz, perfekte Codes, Warum perfekte Codes schlecht sind, Warum Distanz nicht alles ist
12. Nachrichtenaustausch: verteiltes Zählen, Pfadzählen, günstigster Pfad, Minimumsummenalgorithmus
13. Exakte Marginalisierung in Graphen: Faktorgraph, Summenproduktalgorithmus
14. LDPC-Codes: Dichteevolution, Knotenordnung, reguläre und irreguläre Codes, Graphumfang
15. Verlustbehaftete Quellencodierung: Transformationscodierung und JPEG-Kompression

The students apply Bayesian inference to problems in both communications and everyday's life. The students explain the concept of digital communications by means of source compression and forward-error correction coding.

For the design of communication systems, they use the concepts of entropy and channel capacity.

They calculate these quantities for memoryless sources and channels.

The students proof both the source coding and the channel coding theorem.

The students compare various methods of source coding with respect to compression rate and complexity.

The students apply source compression methods to measure mutual information.

The students factorize multivariate functions, represent them by graphs, and marginalize them with respect to various variables.

The students explain the design of error-correcting codes and the role of minimum distance.

They decode error-correcting codes by means of maximum-likelihood decoding and message passing.

The students apply distributed algorithms to problems in both communications and everyday's life.

The students improve the properties of low-density parity-check codes by widening the girth and/or irregularity in the degree distribution.

The students transform source images into the frequency domain to improve lossy compression.

–

Die Studierenden wenden Bayes'sche Inferenz auf Probleme in der Nachrichtentechnik und im Alltagsleben an.

Die Studierenden erklären die konzeptuelle Trennung von digitaler Übertragung in Quellen- und Kanalcodierung.

Kommunikationssysteme entwerfen sie unter Betrachtung von Entropie und Kanalkapazität.

Sie berechnen diese Größen für gedächtnislose Quellen und Kanäle.

Die Studierenden beweisen sowohl das Quellen- als auch das Kanalcodierungstheorem.

Die Studierenden vergleichen verschiedenartige Quellencodierungsverfahren hinsichtlich Komplexität und Kompressionsrate.

Die Studierenden verwenden Quellencodierverfahren zur Messung von Transinformation.

Die Studierenden faktorisieren Funktionen mehrerer Veränderlicher, stellen diese als Graph dar und marginalisieren sie bezüglich mehrerer Veränderlicher.

Die Studierenden erklären den Entwurf von Kanalcodes und den Einfluss der Minimaldistanz.

Sie decodieren Kanalcodes gemäß maximaler Likelihood und Nachrichtenaustausch.

Die Studierenden wenden verteilte Algorithmen auf Probleme der Nachrichtentechnik und des Alltagslebens an.

Die Studierenden verbessern die Eigenschaften von LDPC-Codes durch Erhöhung des Umfangs und/oder durch irreguläre Knotenordnungsverteilungen.

Die Studierenden transformieren Bildquellen zur Verbesserung verlustbehafteter Kompression in den Frequenzbereich.

Literatur:

MacKay, D.: Information Theory, Inference, and Learning Algorithms, Cambridge University Press, Cambridge, 2003.

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

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

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern " Communications Engineering (Master of Science)", "Berufspädagogik Technik (Bachelor of Science)", "Berufspädagogik Technik (Master of Education)", "Communications and Multimedia Engineering (Master of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Elektrotechnik, Elektronik und Informationstechnik (Bachelor of Science)", "Elektrotechnik, Elektronik und Informationstechnik (Master of Science)", "Informatik (Bachelor of Science)", "Informatik (Master of Science)", "Informations- und Kommunikationstechnik (Bachelor of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Mathematik (Bachelor of Science)", "Wirtschaftsingenieurwesen (Bachelor of Science)", "Wirtschaftsingenieurwesen (Master of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Information Theory and Coding / Informationstheorie und Codierung (Prüfungsnummer: 36011)

(englische Bezeichnung: Information Theory and Coding)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 0% Prüfungssprache: Deutsch oder Englisch

Erstablingung: WS 2018/2019, 1. Wdh.: SS 2019

1. Prüfer: Ralf Müller

Bemerkungen:

Schlüsselwörter: ASC

Modulbezeichnung:	Künstliche Intelligenz I (KI I) (Artificial Intelligence I)	7.5 ECTS
Modulverantwortliche/r:	Michael Kohlhase	
Lehrende:	Michael Kohlhase	
Startsemester:	WS 2018/2019	Dauer: 1 Semester
Präsenzzeit:	90 Std.	Eigenstudium: 135 Std.
		Turnus: jährlich (WS)
		Sprache: Deutsch und Englisch
Lehrveranstaltungen:	Künstliche Intelligenz I (WS 2018/2019, Vorlesung, 4 SWS, Michael Kohlhase)	
	Übungen zu Künstliche Intelligenz I (WS 2018/2019, Übung, 2 SWS, Dennis Müller)	

Inhalt:

Dieser Kurs beschäftigt sich mit den Grundlagen der Künstlichen Intelligenz (KI), insbesondere formale Wissensrepräsentation, Heuristische Suche, Automatisches Planen und Schliessen unter Unsicherheit.

—
This course covers the foundations of Artificial Intelligence (AI), in particular symbolic techniques based on search and inference. Lernziele und Kompetenzen:

- Wissen: Die Studierenden lernen grundlegende Repräsentationsformalismen und Algorithmen der Künstlichen Intelligenz kennen.
- Anwenden: Die Konzepte werden an Beispielen aus der realen Welt angewandt (Übungsaufgaben).
- Analyse: Die Studierenden lernen die über die modellierung in der Maschine menschliche Intelligenzleistungen besser einzuschätzen.

Sozialkompetenz

- Die Studierenden arbeiten in Kleingruppen zusammen um kleine Projekte zu bewältigen

Learning Goals and Competencies

Technical, Learning, and Method Competencies

- Knowledge: The students learn foundational representations and algorithms in AI.
- Application: The concepts learned are applied to examples from the real world (homeworks).
- Analysis: By modeling human cognitive abilities, students learn to assess and understand human intelligence better.
- Social Competences: Students work in small groups to solve an AI game-play challenge/competition (Kalah).

Literatur:

Die Vorlesung folgt weitgehend dem Buch

Stuart Russell und Peter Norvig: Artificial Intelligence: A Modern Approach. Prentice Hall, 3rd edition, 2009.

Deutsche Ausgabe:

Stuart Russell und Peter Norvig: Künstliche Intelligenz: Ein Moderner Ansatz. Pearson-Studium, 2004 (Übersetzung der 2.​ Auflage). ISBN: 978-3-8273-7089-1.

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

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

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Bachelor of Arts (2 Fächer))", "Informatik (Bachelor of Science)", "Informatik (Master of Science)", "Mathematik (Bachelor of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Künstliche Intelligenz I (Prüfungsnummer: 535405)

(englische Bezeichnung: Artificial Intelligence I)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100% Prüfungssprache: Englisch

Erstablingung: WS 2018/2019, 1. Wdh.: SS 2019

1. Prüfer: Michael Kohlhase

Modulbezeichnung: Maschinelles Lernen für Zeitreihen Deluxe (MLTS+) 7.5 ECTS
 (Machine Learning for Time Series)

Modulverantwortliche/r: Björn Eskofier, Oliver Amft

Lehrende: Oliver Amft, Björn Eskofier

Startsemester: WS 2018/2019 Dauer: 1 Semester Turnus: jährlich (WS)

Präsenzzeit: 90 Std. Eigenstudium: 135 Std. Sprache: Englisch

Lehrveranstaltungen:

Maschinelles Lernen für Zeitreihen (WS 2018/2019, Vorlesung, 2 SWS, Björn Eskofier et al.)
 Maschinelles Lernen für Zeitreihen Übung (WS 2018/2019, Übung, 2 SWS, Nooshin Haji Ghassemi et al.)
 Maschinelles Lernen für Zeitreihen Laborprojekt (WS 2018/2019, Praktikum, 2 SWS, Nooshin Haji Ghassemi)

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

Pattern Analysis
 Introduction to Pattern Recognition
 Pattern Recognition

Inhalt:

Die Vorlesung vermittelt Konzepte des Maschinellen Lernens speziell im Hinblick auf Anwendungen bei Zeitreihen. Es handelt sich hier um eine Spezialisierungsvorlesung, eine erfolgreiche Absolvierung der Vorlesungen „IntroPR“ und/oder „Pattern Recognition“/„Pattern Analysis“ wird empfohlen. Konzepte, die in „IntroPR“ vermittelt werden, werden hier als Grundwissen vorausgesetzt. Die folgenden Themen werden in der Vorlesung behandelt:

- Ein Überblick über die Anwendungsgebiete der Zeitreihenanalyse
- Methodische Grundlagen des Maschinellen Lernens (ML) für die Analyse von Zeitreihen, beispielsweise Gauß-Prozesse, Monte-Carlo Sampling und Deep Learning
- Design, Implementierung und Evaluation von ML Methoden, um Probleme in Zeitreihen zu adressieren
- Arbeitstechniken in bekannten Toolboxes zur Implementierung von relevanten Methoden, beispielsweise Tensorflow/Keras

Content

Aim of the lecture is to teach Machine learning (ML) methods for a variety of time series applications. The following topics will be covered:

- An overview of applications of time series analysis
- Fundamentals of Machine learning (ML) methods, such as Gaussian processes, Monte Carlo sampling methods and deep learning, for time series analysis
- Design, implementation and evaluation of ML methods in order to address time series problems
- Working with widely-used toolboxes that can be used for implementation of ML methods, such as Tensorflow or Keras

Lernziele und Kompetenzen:

- Die Studierenden sollen ein Verständnis für Zeitreihenprobleme und deren Lösungen in Applikationsgebieten der Industrie, Medizin, dem Finanzwesen, etc. entwickeln
- Die Studierenden erlernen Konzepte des Maschinellen Lernens im Allgemeinen und deren Anwendung auf Zeitreihen im Besonderen

- Die Studierenden erlernen die Charakteristika von Zeitreihendaten und werden zur Entwicklung und Implementierung von ML-Methoden angeleitet, um solche Daten in konkreten Fragestellungen zu modellieren, manipulieren und vorherzusagen.

Learning Objectives

- Students develop an understanding of concepts of time series problems and their wide applications in industry, medicine, finance, etc.

Students learn concepts of machine learning (ML) methods in general and tackling time series problems in particular

- Students understand the characteristics of time series data and will be capable of developing and implementing ML methods to model, predict and manipulate such data in concrete problems
 Studon: https://www.studon.fau.de/studon/goto.php?target=crs_2202195 Literatur:
- Machine Learning: A Probabilistic Perspective, Kevin Murphy, MIT press, 2012
- The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Trevor Hastie, Robert Tibshirani, Jerome Friedman, Springer, 2009
- Deep Learning, Ian Goodfellow and Yoshua Bengio and Aaron Courville, MIT Press, 2016
- Reinforcement Learning: An Introduction, Richard S. Sutton and Andrew G. Barto, MIT press, 1998

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

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

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Elektrotechnik, Elektronik und Informationstechnik (Master of Science)", "Informatik (Bachelor of Arts (2 Fächer))", "Informatik (Bachelor of Science)", "Informatik (Master of Science)", "Mathematik (Bachelor of Science)", "Medizintechnik (Bachelor of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Maschinelles Lernen für Zeitreihen Deluxe (Prüfungsnummer: 482355)

(englische Bezeichnung: Machine Learning for Time Series Deluxe)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100% Prüfungssprache: Deutsch oder Englisch

Erstablesung: WS 2018/2019, 1. Wdh.: SS 2019

1. Prüfer: Björn Eskofier

Modulbezeichnung: Konzeptionelle Modellierung (KonzMod) 5 ECTS
 (Conceptual Modelling)

Modulverantwortliche/r: Richard Lenz

Lehrende: Richard Lenz

Startsemester: WS 2018/2019 Dauer: 1 Semester Turnus: jährlich (WS)

Präsenzzeit: 60 Std. Eigenstudium: 90 Std. Sprache: Deutsch

Lehrveranstaltungen:

Ausschlussbedingung: Wer dieses Modul ablegt, darf das Modul DBNF nicht mehr ablegen.

Konzeptionelle Modellierung (WS 2018/2019, Vorlesung, 2 SWS, Richard Lenz)

Übungen zu Konzeptionelle Modellierung (WS 2018/2019, Übung, 2 SWS, Andreas Maximilian Wahl)

Empfohlene Voraussetzungen:

Gewünscht "Algorithmen und Datenstrukturen" und "Grundlagen der Logik und Logikprogrammierung"

Inhalt:

- Grundlagen der Modellierung

- Datenmodellierung am Beispiel Entity-Relationship-Modell
- Modellierung objektorientierter Systeme am Beispiel UML
- Relationale Datenmodellierung und Anfragemöglichkeiten
- Grundlagen der Metamodellierung
- XML
- Multidimensionale Datenmodellierung
- Domänenmodellierung und Ontologien

Lernziele und Kompetenzen: Die

Studierenden:

- definieren grundlegende Begriffe aus der Datenbankfachliteratur
- erklären die Vorteile von Datenbanksystemen
- erklären die verschiedenen Phasen des Datenbankentwurfs
- benutzen das Entity-Relationship Modell und das erweiterte Entity-Relationship Modell zur semantischen Datenmodellierung
- unterscheiden verschiedene Notationen für ER-Diagramme
- erläutern die grundlegenden Konzepte des relationalen Datenmodells
- bilden ein gegebenes EER-Diagramm auf ein relationales Datenbankschema ab
- erklären die Normalformen 1NF, 2NF, 3NF, BCNF und 4NF
- definieren die Operationen der Relationenalgebra
- erstellen Datenbanktabellen mit Hilfe von SQL
- lösen Aufgaben zur Datenselektion und Datenmanipulation mit Hilfe von SQL
- erklären die grundlegenden Konzepte der XML
- erstellen DTDs für XML-Dokumente
- benutzen XPATH zur Formulierung von Anfragen an XML-Dokumente
- definieren die grundlegenden Strukturelemente und Operatoren des multidimensionalen Datenmodells
- erklären Star- und Snowflake-Schema
- benutzen einfache UML Use-Case Diagramme
- benutzen einfache UML-Aktivitätsdiagramme
- erstellen UML-Sequenzdiagramme
- erstellen einfache UML-Klassendiagramme
- erklären den Begriff Meta-Modellierung
- definieren den Begriff der Ontologie in der Informatik
- definieren die Begriffe RDF und OWL

Literatur:

- Alfons Kemper, Andre Eickler: Datenbanksysteme : Eine Einführung. 6., aktualis. u. erw. Aufl. Oldenbourg, März 2006. - ISBN-10: 3486576909
- Bernd Oestereich: Analyse und Design mit UML 2.1. 8. Aufl. Oldenbourg, Januar 2006. - ISBN-10: 3486579266
- Ian Sommerville: Software Engineering. 8., aktualis. Aufl. Pearson Studium, Mai 2007. - ISBN-10: 3827372577
- Horst A. Neumann: Objektorientierte Softwareentwicklung mit der Unified Modeling Language. (UML). Hanser Fachbuch, März 2002. - ISBN-10: 3446188797
- Rainer Eckstein, Silke Eckstein: XML und Datenmodellierung. Dpunkt Verlag, November 2003. ISBN-10: 3898642224

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

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

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "079#72#H", "079#74#H", "Berufspädagogik Technik (Bachelor of Science)", "Digitale Geistes- und Sozialwissenschaften (Bachelor of Arts (2 Fächer))", "Informatik (1. Staatsprüfung für das Lehramt an Gymnasien)", "Informatik (1. Staatsprüfung für das Lehramt an Realschulen)", "Informatik (Bachelor of Arts (2 Fächer))", "Informatik (Bachelor of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "International Information Systems (IIS) (Master of Science)", "Maschinenbau (Master of Science)", "Mathematik (Bachelor of Science)", "Physische Geographie (Bachelor of Science)", "Technomathematik (Bachelor of Science)", "Wirtschaftsinformatik (Bachelor of Science)", "Wirtschaftsingenieurwesen (Master of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Konzeptionelle Modellierung (Klausur) (Prüfungsnummer: 31301)

(englische Bezeichnung: Written examination in conceptual modelling)

Prüfungsleistung, Klausur mit MultipleChoice, Dauer (in Minuten): 90 Anteil
 an der Berechnung der Modulnote: 100%

Erstablingung: WS 2018/2019, 1. Wdh.: SS 2019

1. Prüfer: Richard Lenz

Modulbezeichnung:	Computergraphik-VUP (CG-VUP) (Computer Graphics)	7.5 ECTS
Modulverantwortliche/r:	Marc Stamminger	
Lehrende:	Marc Stamminger	
Startsemester: WS 2018/2019	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 90 Std.	Eigenstudium: 135 Std.	Sprache: Englisch

Lehrveranstaltungen:

Computergraphik (WS 2018/2019, Vorlesung, 3 SWS, Marc Stamminger)

Übungen zur Computergraphik (WS 2018/2019, Übung, 1 SWS, N.N.)

Vertiefte Übungen zur Computergraphik (WS 2018/2019, Praktikum, 2 SWS, N.N.)

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

Inhalt:

Die Vorlesung gibt eine Einführung in die Computergraphik:

- Graphik Pipeline
- Clipping
- 3D Transformationen
- Hierarchische Display Strukturen
- Perspektive und Projektionen
- Visibilitätsbetrachtungen
- Rastergraphik und Scankonvertierung
- Farbmodelle
- Lokale und globale Beleuchtungsmodelle
- Schattierungsverfahren
- Ray Tracing und Radiosity
- Schatten und Texturen

Lernziele und Kompetenzen: Die
 Studierenden

- geben die unterschiedlichen Schritte der Graphik Pipeline wieder
- erklären die Funktionsweise der Clippingalgorithmen für Linien und Polygone
- beschreiben, charakterisieren und berechnen affine und perspektivische Transformationen in 3D und veranschaulichen die allgemeine Form der Transformationsmatrix in homogener Koordinaten
- skizzieren die Verfahren zur Tiefe- und Visibilitätsberechnung
- vergleichen die unterschiedlichen Farbmodelle der Computergraphik
- illustrieren und untersuchen die Datenstrukturen zur Beschreibung virtueller 3D Modelle und komplexer Szenen
- erläutern die Funktionsweise der Rasterisierung und Scankonvertierung in der Graphikpipeline
- implementieren 3D Transformationen mithilfe der Programmiersprache C++ und der graphischen Bibliothek OpenGL
- Implementieren Beleuchtungsmodelle und Texturierung von virtuellen 3D Objekten mithilfe der Programmiersprachen OpenGL und GLSL
- lösen Aufgaben zu Beleuchtung und Texturierung von 3D virtuellen Modellen
- klassifizieren Schattierungsverfahren
- bestimmen den Unterschied zwischen lokaler und globaler Beleuchtung und formulieren Algorithmen für Ray Tracing und Radiosity

Educational objectives and skills:

Students should be able to

- describe the processing steps in the graphics pipeline
 - explain clipping algorithms for lines and polygons
explain, characterize and compute affine and perspective transformations in 2D and 3D, and provide an intuitive description of the general form of corresponding transformation matrices in homogeneous coordinates
 - depict techniques to compute depth, occlusion and visibility
 - compare the different color models
 - describe data structures to represent 3D virtual models and complex scenes
 - explain the algorithms for rasterization and scan conversion
 - solve problems with shading and texturing of 3D virtual models
 - classify different shadowing techniques
 - explain the difference between local and global illumination techniques and formulate algorithms for ray tracing and radiosity
- Literatur:
- P. Shirley: Fundamentals of Computer Graphics. AK Peters Ltd., 2002
 - Hearn, M. P. Baker: Computer Graphics with OpenGLD. Pearson
 - Foley, van Dam, Feiner, Hughes: Computer Graphics - Principles and Practice
 - Rauber: Algorithmen der Computergraphik
 - Bungartz, Griebel, Zenger: Einführung in die Computergraphik
 - Encarnaçã, Strasser, Klein: Computer Graphics
-

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

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

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Bachelor of Arts (2 Fächer))", "Informatik (Bachelor of Science)", "Informatik (Master of Science)", "Mathematik (Bachelor of Science)", "Medizintechnik (Bachelor of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Computergraphik (Vorlesung mit Übung und Praktikum) (Prüfungsnummer: 321743)

Prüfungsleistung, Klausur, Dauer (in Minuten): 60 Anteil
an der Berechnung der Modulnote: 100% weitere
Erläuterungen:

Übung: 50% der schriftlichen Aufgaben, Praktikum: 50% der Programmieraufgaben, Modulnote
durch Klausur über 60 Minuten

Erstablingung: WS 2018/2019, 1. Wdh.: SS 2019

1. Prüfer: Marc Stamminger

Bemerkungen:

Vorlesungsunterlagen, Übungsblätter und die Klausur sind in englischer Sprache

Modulbezeichnung: Deep Learning (DL) 5 ECTS
(Deep Learning)

Modulverantwortliche/r: Andreas Maier

Lehrende: Andreas Maier, Vincent Christlein, Lennart Husvogt, Tobias Würfl

Startsemester: WS 2018/2019 Dauer: 1 Semester Turnus: jährlich (WS)

Präsenzzeit: 60 Std. Eigenstudium: 90 Std. Sprache: Englisch

Lehrveranstaltungen:

Deep Learning (WS 2018/2019, Vorlesung, 2 SWS, Andreas Maier)

Deep Learning Exercises (WS 2018/2019, Übung, 2 SWS, Leonid Mill et al.)

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

Pattern Recognition Deluxe

Introduction to Pattern Recognition Deluxe

Inhalt:

Deep Learning (DL) has attracted much interest in a wide range of applications such as image recognition, speech recognition and artificial intelligence, both from academia and industry. This lecture introduces the core elements of neural networks and deep learning, it comprises:

- (multilayer) perceptron, backpropagation, fully connected neural networks
- loss functions and optimization strategies
- convolutional neural networks (CNNs)
- activation functions
- regularization strategies
- common practices for training and evaluating neural networks
- visualization of networks and results
- common architectures, such as LeNet, Alexnet, VGG, GoogleNet
- recurrent neural networks (RNN, TBPTT, LSTM, GRU)
- deep reinforcement learning
- unsupervised learning (autoencoder, RBM, DBM, VAE)
- generative adversarial networks (GANs)
- weakly supervised learning
- applications of deep learning (segmentation, object detection, speech recognition, ...)

The accompanying exercises will provide a deeper understanding of the workings and architecture of neural networks.

Lernziele und Kompetenzen:

The students

- explain the different neural network components,
 - compare and analyze methods for optimization and regularization of neural networks,
 - compare and analyze different CNN architectures,
 - explain deep learning techniques for unsupervised / semi-supervised and weakly supervised learning,
 - explain deep reinforcement learning,
 - explain different deep learning applications,
 - implement the presented methods in Python,
 - autonomously design deep learning techniques and prototypically implement them,
 - effectively investigate raw data, intermediate results and results of Deep Learning techniques on a computer,
 - autonomously supplement the mathematical foundations of the presented methods by self-guided study of the literature,
 - discuss the social impact of applications of deep learning applications. Literatur:
 - Ian Goodfellow, Yoshua Bengio, Aaron Courville: Deep Learning. MIT Press, 2016.
 - Christopher Bishop: Pattern Recognition and Machine Learning, Springer Verlag, Heidelberg, 2006
 - Yann LeCun, Yoshua Bengio, Geoffrey Hinton: Deep learning. Nature 521, 436 - 444 (28 May 2015)
-

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

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

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Deep Learning (Prüfungsnummer: 901895)

(englische Bezeichnung: Deep Learning)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 0% Prüfungssprache: Deutsch oder Englisch

Erstabelleung: WS 2018/2019, 1. Wdh.: SS 2019

1. Prüfer: Andreas Maier

Modulbezeichnung:	Eingebettete Systeme mit erweiterter Übung (ES-VEU) (Embedded Systems with Extended Exercises)	7.5 ECTS
Modulverantwortliche/r:	Jürgen Teich	
Lehrende:	Jürgen Teich, Frank Hannig	
Startsemester: WS 2018/2019	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 90 Std.	Eigenstudium: 135 Std.	Sprache: Deutsch
Lehrveranstaltungen:		
Eingebettete Systeme (WS 2018/2019, Vorlesung, 2 SWS, Jürgen Teich et al.)		
Übung zu Eingebettete Systeme (WS 2018/2019, Übung, 2 SWS, Jörg Fickenscher et al.)		
Erweiterte Übungen zu Eingebettete Systeme (WS 2018/2019, Übung, Jörg Fickenscher et al.)		

Inhalt:

Schwerpunkt der Vorlesung ist der Entwurf und die Implementierung eingebetteter Systeme unter Einsatz formaler Methoden und rechnergestützter Entwurfsverfahren.

Unter eingebetteten Systemen versteht man Rechensysteme, die auf einen Anwendungsbereich zugeschnitten (z.B. mobile Kommunikationsgeräte, Chipkartensysteme, Industriesteuerungen, Unterhaltungselektronik, Medizintechnik) und in einen technischen Kontext eingebunden sind. Das große Interesse am systematischen Entwurf von heterogenen eingebetteten Systemen ist verursacht durch die steigende Vielfalt und Komplexität von Anwendungen für eingebettete Systeme, die Notwendigkeit, Entwurfs- und Testkosten zu senken sowie durch Fortschritte in Schlüsseltechnologien (Mikroelektronik, formale Methoden).

The focus of the lecture is the design and implementation of embedded systems using formal methods and computer-aided design techniques.

Embedded systems are computing systems tailored for a particular application (e.g., mobile communication devices, smart card systems, industrial control, consumer electronics, medical technology) and integrated into a technical context. The keen interest in the systematic design of heterogeneous embedded systems is driven by the increasing diversity and complexity of embedded system applications, the need to reduce design and test costs, and advances in key technologies (microelectronics, formal methods).

Lernziele und Kompetenzen:

Fachkompetenz

Wissen

- Die Studierenden setzen sich mit einem aktuellen Forschungsgebiet auseinander. / *The students deal with a current field of research.*
- Die Studierenden lernen aktuelle Entwurfswerkzeuge für die Architektursynthese (Hardware) und Softwaresynthese kennen. / *The students learn about current design tools for architectural synthesis (hardware) and software synthesis.*

Verstehen

- Die Studierenden verstehen grundlegende Konzepte des Entwurfs eingebetteter Systeme. / *The students become familiar with the fundamental concepts of designing of embedded systems.*

Anwenden

- Die Studierenden wenden grundlegende Algorithmen an zur Analyse und Optimierung von Hardware-Architekturen und Echtzeit-Softwaresystemen. / *The students apply basic algorithms to analyze and optimize hardware architectures and real-time software systems.*
- Die Studierenden erfassen den Hardware/Software-Entwurf von Systemen mit harten Beschränkungen. / *The students understand the hardware/software design of hard-constrained systems. Sozialkompetenz*

- Die Studierenden lernen aktuelle Entwurfswerkzeuge für die Architektursynthese (Hardware) und Softwaresynthese kennen bei der kooperativen Bearbeitung der erweiterten Übung in Gruppen. / *The students learn about current design tools for architectural synthesis (hardware) and software synthesis by processing the extended exercises in groups cooperatively.*

Literatur:

- Buch zur Vorlesung
 - Vorlesungsskript (Zugriff nur innerhalb des Uni-Netzwerks möglich)
-

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

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

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Bachelor of Arts (2 Fächer))", "Informatik (Bachelor of Science)", "Informatik (Master of Science)", "Informations- und Kommunikationstechnik (Bachelor of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Mathematik (Bachelor of Science)", "Medizintechnik (Bachelor of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Eingebettete Systeme (Vorlesung mit erweiterten Übungen) (Prüfungsnummer: 209679)

Prüfungsleistung, mehrteilige Prüfung

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

Portfolio besteht aus: Klausur (90 min) und erfolgreicher Teilnahme an den erweiterten Übungen (verpflichtend)

Die Modulnote ergibt sich aus der Klausurnote.

Erstablingung: WS 2018/2019, 1. Wdh.: SS 2019

1. Prüfer: Jürgen Teich

Organisatorisches:

Die Auswahl dieses Moduls schließt die Auswahl des Moduls „Eingebettete Systeme“ aus.

Modulbezeichnung:	Ereignisgesteuerte Systeme (EGS) (Discrete Event Systems)	5 ECTS
Modulverantwortliche/r:	Stefan Wildermann	
Lehrende:	Stefan Wildermann	
Startsemester: WS 2018/2019	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

Lehrveranstaltungen:

Ereignisgesteuerte Systeme (WS 2018/2019, Vorlesung, 2 SWS, Stefan Wildermann)

Übung zu Ereignisgesteuerte Systeme (WS 2018/2019, Übung, 2 SWS, Stefan Wildermann et al.)

Inhalt:

Die rasante Entwicklung von Rechnertechnologien in den vergangenen Jahrzehnten hatte die Verbreitung neuer dynamischer und komplexer Systeme zu Folge. Wesentliche Charakteristika solcher Systeme sind Verteiltheit, Nebenläufigkeit und das asynchrone Auftreten diskreter Ereignisse. Der Prozess, neue Modelle und Methoden für ereignisgesteuerte Systeme zu entwickeln, ist vergleichsweise jung. Der Rechner selbst spielt hierbei eine entscheidende Rolle als Werkzeug für Systementwurf, Analyse und Steuerung. Die Vorlesung EGS hat zum Ziel, Modellierungs-, Simulations- und Entwurfsmethoden für verteilte und ereignisdiskrete Systeme zu vermitteln. Die Methoden werden dabei beispielhaft auf Anwendung aus den Bereichen Computernetzwerke, automatischen

Produktionssysteme, komplexen Softwaresysteme und integrierte Steuerungs-, Kommunikations- und Informationssysteme angewendet. In diesem Kontext behandelt die Vorlesung daher die folgenden Themenbereiche:

- Eigenschaften komplexer Systeme
- Überblick über Systeme und Modelle
- Zeitfreie und zeitbehaftete Modelle
- Stochastische Modelle
- Umsetzung in Programmiersprachen
- Simulation-, Entwurfs- und Testverfahren auf der Basis der vorgestellten Modelle.

Lernziele und Kompetenzen:

Fachkompetenz

Verstehen

- Die Studierenden erläutern grundlegende Techniken zur Modellierung diskreter, ereignisgesteuerter Systeme, zeigen deren Vor- und Nachteile auf und vergleichen diese bezüglich Ihrer Mächtigkeit.

Anwenden

- Die Studierenden wenden Modellierungs- und Analysetechniken aus dem Bereich endlicher Automaten, Petri-Netze, Markov-Ketten auf komplexe Systeme an.
 - Die Studierenden setzen die Modellierung und Analyse eines Systems mit einem konkreten Entwurfswerkzeug um.
-

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

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

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Bachelor of Arts (2 Fächer))", "Informatik (Bachelor of Science)", "Informatik (Master of Science)", "Informations- und Kommunikationstechnik (Bachelor of Science)", "Mathematik (Bachelor of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Vorlesung und Übung Ereignisgesteuerte Systeme (Prüfungsnummer: 35401)

(englische Bezeichnung: Lecture/Tutorial: Discrete Event Systems)

Prüfungsleistung, Klausur

Anteil an der Berechnung der Modulnote: 100%

Erstablingung: WS 2018/2019, 1. Wdh.: SS 2019

1. Prüfer: Jürgen Teich

Bemerkungen:

auch für Computational Engineering und I&K

Modulbezeichnung:	Systemprogrammierung Vertiefung (VSP) (Advanced Systems Programming)	5 ECTS
Modulverantwortliche/r:	Wolfgang Schröder-Preikschat	
Lehrende:	Wolfgang Schröder-Preikschat, Jürgen Kleinöder	

Startsemester: WS 2018/2019	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

Lehrveranstaltungen:

- Systemprogrammierung 2 (WS 2018/2019, Vorlesung, 2 SWS, Jürgen Kleinöder et al.)
 - Übungen zu Systemprogrammierung 2 (WS 2018/2019, Übung, 2 SWS, Christian Eichler et al.)
 - Rechnerübungen zu Systemprogrammierung 1 und 2 (WS 2018/2019, Übung, 2 SWS, Christian Eichler et al.)
-

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

Inhalt:

- Grundlagen von Betriebssystemen (Adressräume, Speicher, Dateien, Prozesse, Koordinationsmittel; Betriebsarten, Einplanung, Einlastung, Virtualisierung, Nebenläufigkeit, Koordination/Synchronisation)
- Abstraktionen/Funktionen UNIX-ähnlicher Betriebssysteme
- Programmierung von Systemsoftware

Lernziele und Kompetenzen: Die Studierenden

- erwerben fundierte Kenntnisse über Grundlagen von Betriebssystemen
 - verstehen Zusammenhänge, die die Ausführungen von Programmen in vielschichtig organisierten Rechensystemen ermöglichen
 - erkennen Gemeinsamkeiten und Unterschiede zwischen realen und abstrakten (virtuellen) Maschinen • erlernen die Programmiersprache C
 - entwickeln Systemprogramme auf Basis der Systemaufrufschnittstelle UNIX-ähnlicher Betriebssysteme
-

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

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

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "Mathematik (Master of Science)", "Medizintechnik (Bachelor of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Systemprogrammierung Vertiefung (Prüfungsnummer: 650143)

(englische Bezeichnung: Advanced Systems Programming)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 20

Anteil an der Berechnung der Modulnote: 0% weitere

Erläuterungen:

Erfolgreiche Bearbeitung der Übungsaufgaben (6 Programmieraufgaben, es müssen mind. 50 % der insgesamt erreichbaren Punkte erreicht werden), Vorstellung mind. einer Aufgabenbearbeitung in einer Tafelübung.

Mündliche Prüfung (ca. 20 min.) über den Stoff von Vorlesung und Übungen. Die Modulnote ergibt sich zu 100 % aus der Note der mündlichen Prüfung.

Erstabwegung: WS 2018/2019, 1. Wdh.: SS 2019

1. Prüfer: Jürgen Kleinöder

Modulbezeichnung:	Algorithms of Numerical Linear Algebra (ANLA) (Algorithms of Numerical Linear Algebra)	7.5 ECTS
Modulverantwortliche/r:	Ulrich Rüde	
Lehrende:	Ulrich Rüde	

Startsemester: WS 2018/2019	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 165 Std.	Sprache: Englisch

Lehrveranstaltungen:

Algorithms of Numerical Linear Algebra (WS 2018/2019, Vorlesung mit Übung, 4 SWS, Ulrich Rüde)
Exercises in Algorithms of Numerical Linear Algebra (WS 2018/2019, Übung, 2 SWS, Sebastian Eibl)

Empfohlene Voraussetzungen:

- Elementary Numerical Mathematics
 - Engineering Mathematics or Equivalent,
-

Inhalt:

- Vectors
- Matrices
- Vector Spaces
- Matrix Factorizations
- Orthogonalisation
- Singular Value Decomposition
- Eigenvalues
- Krylov Space Methods
- Arnoldi Method
- Lanczos Method
- Multigrid

Lernziele und Kompetenzen:

Providing students with a solid theoretical background for the foundations of modern solution techniques in Computational Engineering Literatur:

Trefethen, Bau: Numerical Linear Algebra, SIAM 1997

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

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

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Informatik (Master of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Algorithms of Numerical Linear Algebra (Prüfungsnummer: 352989)

(englische Bezeichnung: Algorithms of Numerical Linear Algebra)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90 Anteil

an der Berechnung der Modulnote: 100% weitere

Erläuterungen:

Die Note ergibt sich aus einer mündlichen Prüfung. Bei mehr als 10 Teilnehmern kann die Prüfung als 90 minütige Klausur durchgeführt werden. Voraussetzung zur Teilnahme an der Prüfung ist die erfolgreiche Bearbeitung der Übungsaufgaben.

Erstablingung: WS 2018/2019, 1. Wdh.: SS 2019, 2. Wdh.: WS 2019/2020 1.

Prüfer: Ulrich Rüde

Organisatorisches:

Lectures and Exercises will be mixed in a flexible way

Modulbezeichnung:	Reconfigurable Computing (RC-VU) (Reconfigurable Computing)		5 ECTS
Modulverantwortliche/r:	Jürgen Teich		
Lehrende:	Jürgen Teich		
Startsemester: WS 2018/2019	Dauer: 1 Semester	Turnus: jährlich (WS)	
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Englisch	
Lehrveranstaltungen:			
Reconfigurable Computing (WS 2018/2019, Vorlesung, 2 SWS, Jürgen Teich)			
Exercises to Reconfigurable Computing (WS 2018/2019, Übung, 2 SWS, Faramarz Khosravi et al.)			

Inhalt:

Content:

Reconfigurable (adaptive) computing is a novel yet important research field investigating the capability of hardware to adapt to changing computational requirements such as emerging standards, late design changes, and even to changing processing requirements arising at run-time. Reconfigurable computing thus benefits from a) the programmability of software similar to the Von Neumann computer and b) the speed and efficiency of parallel hardware execution.

The purpose of the course reconfigurable computing is to instruct students about the possibilities and rapidly growing interest in adaptive hardware and corresponding design techniques by providing them the necessary knowledge for understanding and designing reconfigurable hardware systems and studying applications benefiting from dynamic hardware reconfiguration.

After a general introduction about benefits and application ranges of reconfigurable (adaptive) computing in contrast to general-purpose and application-specific computing, the following topics will be covered:

- Reconfigurable computing systems: Introduction of available technology including fine grained look up table (LUT-) based reconfigurable systems such as field programmable gate arrays (FPGA) as well as newest coarse grained architectures and technology.
- Design and implementation: Algorithms and steps (design entry, functional simulation, logic synthesis, technology mapping, place and route, bit stream generation) to implement (map) algorithms to FPGAs. The main focus lies on logic synthesis algorithms for FPGAs, in particular LUT technology mapping.
- Temporal partitioning: techniques to reconfigure systems over time. Covered are the problems of mapping large circuits which do not fit one single device. Several temporal partitioning techniques are studied and compared.
- Temporal placement: Techniques and algorithms to exploit the possibility of partial and dynamic (run-time) hardware reconfiguration. Here, OS-like services are needed that optimize the allocation and scheduling of modules at run-time.
- On-line communication: Modules dynamically placed at run-time on a given device need to communicate as well as transport data off-chip. State-of-the-art techniques are introduced how modules can communicate data at run-time including bus-oriented as well as network-on-a-chip (NoC) approaches.
- Designing reconfigurable applications on Xilinx Virtex FPGAs: In this part, the generation of partial bitstreams for components to be placed at run-time on Xilinx FPGAs is introduced and discussed including newest available tool flows.
- Applications: This section presents applications benefiting from dynamic hardware reconfiguration. It covers the use of reconfigurable systems including rapid prototyping, reconfigurable supercomputers, reconfigurable massively parallel computers and studies important application

domains such as distributed arithmetic, signal processing, network packet processing, control design, and cryptography.

Lernziele und Kompetenzen:

Learning objectives and competencies:

Fachkompetenz

Wissen

- The students know to exploit run-time reconfigurable design methodologies for adaptive applications.

Verstehen

- The students understand the mapping steps, and optimization algorithms.
- The students classify different types and kinds of reconfigurable hardware technologies available today.
- The students clarify pros and cons of reconfigurable computing technology.
- The students summarize applications benefiting from reconfigurable computing.
- The students describe the design of circuits and systems-on-a-chip (SoC) on FPGAs.

Literatur:

- The Hamburg VHDL Archive (see Documentation link for free books)
 - Interactive VHDL Tutorial with 150 examples from ALDEC
 - Easy FPGA tutorials, projects and boards
 - Xilinx WebPack ISE and Modelsim MXE (free FPGA synthesis tool and free VHDL simulator)
 - Symphone EDA free VHDL simulator (select FREE Edition license)
 - Icarus open-source Verilog simulator
-

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

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

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern " Communications Engineering (Master of Science)", "Informatik (Bachelor of Arts (2 Fächer))", "Informatik (Bachelor of Science)", "Informatik (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Mathematik (Bachelor of Science)", "Mechatronik (Bachelor of Science)", "Mechatronik (Master of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Reconfigurable Computing (Prüfungsnummer: 31951)

(englische Bezeichnung: Reconfigurable Computing)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100% weitere

Erläuterungen:

Oral examination (Duration: 30 min) + Successful completion of all tasks in the exercises (mandatory)

Final grade of the module is determined by the oral examination.

Erstablingung: WS 2018/2019, 1. Wdh.: SS 2019

1. Prüfer: Jürgen Teich

Organisatorisches:

Selection of this module prohibits the selection of the module "Reconfigurable Computing with Extended Exercises (RC-VEU)" by the student.

Bemerkungen:

Reconfigurable computing is an interdisciplinary field of research between computer science and electrical engineering on a 4 SWS (4 hours/week) basis. Lecture and Exercises will give 5 ECTS, the FPGA & VHDL labs 2.5 ECTS.

Modulbezeichnung:	Echtzeitsysteme-V+Ü (EZS-VU) (Real-Time Systems L+E)	5 ECTS
Modulverantwortliche/r:	Peter Ulbrich	
Lehrende:	Peter Ulbrich	
Startsemester: WS 2018/2019	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

Lehrveranstaltungen:

Echtzeitsysteme (WS 2018/2019, Vorlesung, 2 SWS, Peter Ulbrich)
 Übungen zu Echtzeitsysteme (WS 2018/2019, Übung, 2 SWS, Florian Schmaus et al.)
 Rechnerübungen zu Echtzeitsysteme (WS 2018/2019, Übung, Peter Ulbrich)

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

Systemprogrammierung
 Systemnahe Programmierung in C

Inhalt:

Videobearbeitung in Echtzeit, Echtzeitstrategiespiel, echtzeitfähig - der Begriff Echtzeit ist wohl einer der am meisten strapazierten Begriffe der Informatik und wird in den verschiedensten Zusammenhängen benutzt. Diese Vorlesung beschäftigt sich mit dem Begriff Echtzeit aus der Sicht von Betriebssystemen - was versteht man eigentlich unter dem Begriff Echtzeit im Betriebssystemumfeld, wo und warum setzt man sog. Echtzeitbetriebssysteme ein und was zeichnet solche Echtzeitbetriebssysteme aus? In dieser Vorlesung geht es darum, die oben genannten Fragen zu beantworten, indem die grundlegenden Techniken und Mechanismen vermittelt werden, die man im Betriebssystemumfeld verwendet, um Echtzeitsysteme und Echtzeitbetriebssysteme zu realisieren. Im Rahmen dieser Vorlesung werden unter anderem folgende Themen behandelt:

- zeitgesteuerte und ereignisgesteuerte Systeme
- statische und dynamische Ablaufplanungsverfahren
- Fadensynchronisation in Echtzeitbetriebssystemen
- Behandlung von periodischen und nicht-periodischen Ereignissen

In den begleitenden Übungen werden die in der Vorlesung vorgestellten Techniken bei der Entwicklung eines kleinen Echtzeitsystems praktisch umgesetzt.

Lernziele und Kompetenzen:

Studierende, die das Modul erfolgreich abgeschlossen haben:

- unterscheiden die verschiedenen Komponenten eines Echtzeitsystems.
- bewerten die Verbindlichkeiten von Terminvorgaben (weich, fest, hart).
- erläutern die Zusammensetzung des Laufzeitverhaltes einer Echtzeitanwendung.
- klassifizieren die Berührungspunkte zwischen physikalischem Objekt und kontrollierendem Echtzeitsystem.
- interpretieren die Zeitparameter des durch das Echtzeitrechensystem zu kontrollierenden Objekts.
- nennen die Zeitparameter des zugrundeliegenden Rechensystems (Unterbrechungslatenz, Ausführungszeit, ...).
- unterscheiden synchrone und asynchrone Programmunterbrechung (insbesondere Trap/Interrupt, Ausnahmebehandlung und Zustandssicherung).
- skizzieren die Verwaltungsgemeinkosten des schlimmsten Falls.

- entwickeln in der Programmiersprache C und wenden die GNU Werkzeugkette für den ARM Cortex M4 Microcontroller an.
- erstellen Echtzeitanwendungen auf Basis der eCos OS-Schnittstelle
- ordnen die Strukturelemente von Echtzeitanwendungen zu: Aufgabe, Arbeitsauftrag und Faden.
- erläutern die Implikationen von zeitlichem Mehrfachbetrieb auf die Verwaltungsgemeinkosten.
- unterscheiden die Umsetzungsalternativen zur Ablaufsteuerung und die Trennung der Belange in Einplanung (Strategie) und Einlastung (Mechanismus).
- benennen die grundsätzliche Verfahren der Ablaufsteuerung (taktgesteuert, reihum, vorranggesteuert).
- erklären die grundlegenden Zeitparameter einer Aufgabe (Auslösezeitpunkt, Termin, Antwortzeit, Latenz, Ausführungszeit, Schlupfzeit).
- unterscheiden die Grundlagen der Planbarkeit (gültig vs. zulässig, Optimalität von Einplanungsalgorithmen).
- beschreiben den Unterschied zwischen konstruktiver und analytischer Einhaltung von Terminen.
- vergleiche die Möglichkeiten (statisch, dynamisch) der zeitliche Analyse von Echtzeitanwendungen.
- erklären die Grundlagen und Beschränkungen von dynamischer (worst-case?) und statischer WCET-Analyse (makroskopisch und mikroskopisch).
- illustrieren Lösungsverfahren zur Bestimmung des längsten Ausführungspfads (Timing Schema, IPET).
- erstellen Zeitmessung mittels Zeitgeber / Oszilloskop und bestimmen den längsten Pfad durch CodeReview.
- erproben werkzeuggestützte WCET-Analyse mittels des absint aiT Analysewerkzeugs.
- beschreiben die Grundlagen der Abfertigung periodischer Echtzeitsysteme (Periode, Phase, Hyperperiode).
- skizzieren das periodische Modell und dessen Folgen (Entwicklungskomfort vs. Analysierbarkeit).
- erklären die ereignisgesteuerte Ausführung (feste und dynamische Priorität, Verdrängbarkeit) mittels ereignisorientierter Planer (Berechnungskomplexität, MLQ-Scheduler, O(1)-Scheduler).
- unterscheiden die zeitgesteuerte Ausführung (Busy Loop, Ablaufplan) und die Abfertigung von Arbeitsaufträgen im Abfrage- bzw. Unterbrecherbetrieb.
- wenden die Grundlagen der ereignisgesteuerten Ablaufplanung periodischer Echtzeitsysteme an.
- unterscheiden Verfahren zur statischen (RM, DM) und dynamischen Prioritätsvergabe (EDF, LRT, LST).
- nennen den Unterschied zwischen Anwendungs- und Systemebene (Mehrdeutigkeit von Prioritäten).
- erläutern den Optimalitätsnachweis des RM-, DM- und EDF-Algorithmus und dessen Ausnahmen.
- beschreiben grundlegende Verfahren zur Planbarkeitsanalyse (CPU-Auslastung, Antwortzeitanalyse).
- implementieren komplexe Aufgabensysteme in eCos.
- unterscheiden die Grundlagen der zeitgesteuerten Ablaufplanung periodischer Echtzeitsysteme.
- erstellen regelmäßige, zyklische Ablaufpläne (cyclic executive model, Rahmen).
- vergleiche Methoden der manuellen und algorithmischen Ablaufplanung.
- unterscheiden optimale von heuristischen Verfahren (List Scheduling, Branch & Bound).
- diskutieren die Konsequenzen eines Betriebswechsels in Echtzeitsystemen.
- erstellen takt- beziehungsweise ereignisgesteuerte Abläufe in eCos beziehungsweise tt-eCos.
- klassifizieren die Grundlagen der Abfertigung nicht-periodischer Echtzeitsysteme (minimale Zwischenankunftszeit).
- definieren die Verbindlichkeiten von nicht-periodischen Aufgaben (aperiodisch, sporadisch)
- zeigen die sich ergebenden Restriktionen des periodischen Modells (Mischbetrieb, Prioritätswarteschlangen, Übernahmeprüfung) auf.

- beschreiben die Basistechniken des Laufzeitsystems (Zusteller, Unterbrecherbetrieb, Hintergrundbetrieb).
- quantifizieren die Eigenschaften und Auswirkungen auf den periodischen Teil des Echtzeitsystems.
- formulieren die Grundlagen des Slack-Stealing.
- beschreiben den Einsatz von bandweite-bewahrenden Zustellern.
- unterscheiden aufschiebbare Zusteller und Sporadic Server (SpSL und POSIX).
- wenden eine Übernahmeprüfung bei sporadischen Aufgaben mittels dichte- oder schlupfbasierten Akzeptanztests an.
- arbeiten einen strukturierter Ablaufplan (Rahmen) aus und untersuchen den Einsatz von Slack-Stealing.
- ermitteln gerichtete Abhängigkeiten und Rangfolgen in Echtzeitanwendungen (Abhängigkeits- und Aufgabengraph).
- stellen Umsetzungsalternativen für Abhängigkeiten einander gegenüber (naiv, implizit, explizit).
- beschreiben das Konzept der zeitlichen Domänen und physikalischer bzw. logischer Ereignisse.
- übertragen Abhängigkeiten auf das Problem der Ablaufplanung (modifizierte Auslösezeitpunkt/Termin, Phasenversatz).
- konzipieren Rangfolge und aperiodische Steuerung in eCos.
- implementieren einen aperiodischer Moduswechsel mit Zustandsüberführung in eCos.
- wenden die Grundlagen von Wettstreit um Betriebsmitteln, Konkurrenz und Konfliktsituationen (kritische Abschnitte, (un)kontrollierte Prioritätsumkehr) an.
- beschreiben echtzeitfähige Synchronisationsprotokolle (NPCS, PI, PCP).
- nennen die Vor- und Nachteile der Techniken (transitive Blockung, Verklemmungen).
- hinterfragen die Vereinfachung des PCP durch stapelbezogene Grenzprioritäten.
- bestimmen die Ablaufplanung unter Berücksichtigung von Blockierungszeiten und Selbstsuspendierung.
- implementieren Zugriffskontrolle (NPCS, PI, PCP) in Echtzeitanwendungen mit eCos.
- erläutern die Anforderungen an verteilte Echtzeitsysteme (Komposition, Erweiterbarkeit, Komplexität, Ereignis- vs. Zustandsnachricht).
- fassen die Grundlagen von Knoten, Netzwerkschnittstellen und Netzübergängen sowie die Konzepte der expliziten und impliziten Flusskontrolle zusammen.
- erschließen sich typische Probleme (zeitliche Analyse, Beobachtbarkeit, Synchronisation, Rangfolge) und Fehlerquellen bei der Programmierung von Echtzeitanwendungen.
- können in Gruppen kooperativ und effektiv arbeiten.
- können ihre Entwurfs- und Implementierungsentscheidungen kompakt präsentieren und argumentativ vertreten.
- reflektieren ihre Entscheidungen kritisch und leiten Alternativen ab.
- können offen und konstruktiv mit Schwachpunkten in der Konzeption wie Umsetzung umgehen.

Literatur:

- Hermann Kopetz. Real-Time Systems: Design Principles for Distributed Embedded Applications. Kluwer Academic Publishers, 1997.
- Jane W. S. Liu. Real-Time Systems. Prentice-Hall, Inc., 2000.
- Wolfgang Schröder-Preikschat. Softwaresysteme 1. Vorlesungsfolien. 2006.

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

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

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "123#67#H", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Informatik (Bachelor of Arts (2 Fächer))", "Informatik (Bachelor of Science)", "Informatik (Master of Science)", "Informations- und Kommunikationstechnik (Bachelor of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Maschinenbau (Bachelor of Science)", "Maschinenbau (Master of Science)", "Mathematik (Bachelor of Science)", "Mechatronik (Bachelor of Science)", "Mechatronik (Master of Science)", "Medizintechnik (Bachelor of Science)", "Wirtschaftsingenieurwesen (Bachelor of Science)", "Wirtschaftsingenieurwesen (Master of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Mündliche Prüfung Echtzeitsysteme (Prüfungsnummer: 39401)

(englische Bezeichnung: Oral Examination on Real-Time Computing)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100% weitere

Erläuterungen:

30-minütige mündliche Prüfung über den Stoff der Vorlesung und die Übungen.

Teilnahme an den Übungen und die erfolgreiche Bearbeitung aller Übungsaufgaben wird hierzu dringend empfohlen!

Erstabllegung: WS 2018/2019, 1. Wdh.: SS 2019

1. Prüfer: Peter Ulbrich

Modulbezeichnung:	Statistische Signalverarbeitung (STASIP) (Statistical Signal Processing)	5 ECTS
Modulverantwortliche/r:	Walter Kellermann	
Lehrende:	Alexander Schmidt, Walter Kellermann	
Startsemester: WS 2018/2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Englisch

Lehrveranstaltungen:

Statistische Signalverarbeitung (WS 2018/2019, Vorlesung, 3 SWS, Walter Kellermann)
 Ergänzungen und Übungen zur statistischen Signalverarbeitung (WS 2018/2019, Übung, 1 SWS, Alexander Schmidt)

Empfohlene Voraussetzungen:

Module ‚Signale und Systeme I‘ und ‚Signale und Systeme II‘, ‚Digitale Signalverarbeitung‘ oder gleichwertige

Inhalt:

The course concentrates on fundamental methods of statistical signal processing and their applications. The main topics are:

Discrete-time stochastic processes in the time and frequency domain Random variables (RVs), probability distributions and densities, expectations of random variables, transformation of RVs, vectors of normally distributed RVs, time-discrete random processes: probability distribution and densities, expectation, stationarity, cyclostationarity, ergodicity, correlation functions and correlation matrices, spectral representations, principal component analysis (PCA), Karhunen-Loève transform (KLT). Estimation theory estimation criteria, prediction, classical and Bayesian parameter estimation (including MMSE, Maximum Likelihood, and Maximum A Posteriori estimation), Cramer-Rao bound
 Linear signal models

Parametric models (cepstral decomposition, Paley-Wiener theorem, spectral flatness), non-parametric models (all-pole, all-zero and pole-zero models, lattice structures, Yule-Walker equations, PARCOR coefficients, cepstral representation)

Signal estimation

Supervised estimation, problem classes, orthogonality principle, MMSE estimation, linear MMSE estimation for normally distributed random processes, optimum FIR filtering, optimum linear filtering for stationary processes, prediction and smoothing, Kalman filters, optimum multichannel filtering (Wiener filter, LCMV, MVDR, GSC)

Adaptive filtering

Gradient methods, LMS, NLMS, APA and RLS algorithms and their convergence behavior

Zeitdiskrete Zufallsprozesse im Zeit- und Frequenzbereich

Zufallsvariablen (ZVn), Wahrscheinlichkeitsverteilungen und -dichten, Erwartungswerte; Transformation von ZVn; Vektoren normalverteilter ZVn; zeitdiskrete Zufallsprozesse (ZPe): Wahrscheinlichkeitsverteilungen und -dichten, Erwartungswerte, Stationarität, Zyklstationarität, Ergodizität, Korrelationsfunktionen und -matrizen, Spektraldarstellungen; ‚Principal Component Analysis‘, Karhunen-Loeve Transformation;

Schätztheorie

Schätzkriterien; Prädiktion; klassische und Bayes'sche Parameterschätzung (inkl. MMSE, Maximum Likelihood, Maximum A Posteriori); Cramer-Rao-Schranke

Lineare Signalmodelle

Parametrische Modelle (Cepstrale Zerlegung, Paley-Wiener Theorem, Spektrale Glattheit);
Nichtparametrische Modelle: ‚Allpole‘-/‘Allzero‘-/‘Pole-zero‘-(AR/MA/ARMA) Modelle; ‚Lattice‘-
Strukturen, Yule-Walker Gleichungen, PARCOR-Koeffizienten, Cepstraldarstellungen;

Signalschätzung

Überwachte Signalschätzung, Problemklassen; Orthogonalitätsprinzip, MMSE-Schätzung, lineare
MMSE-Schätzung für Gaußprozesse; Optimale FIR-Filter; Lineare Optimalfilter für stationäre Prozesse;
Prädiktion und Glättung; Kalman-Filter; optimale Multikanalfilterung (Wiener-Filter, LCMV, MVDR,
GSC);

Adaptive Filterung

Gradientenverfahren; LMS-, NLMS-, APA- und RLS-Algorithmus und Ihr Konvergenzverhalten; The
course concentrates on fundamental methods of statistical signal processing and their applications.
The main topics are:

Discrete-time stochastic processes in the time and frequency domain Estimation theory Non-
parametric and parametric signal models (pole/zero models, ARMA models) Optimum linear filters
(e.g. for prediction), eigenfilters, Kalman filters Algorithms for optimum linear filter identification
(adaptive filters)

Course material

To be kept up to date, please register for the course on StudOn. Extra points for the written exam
Extra points for the written exam can be obtained by handing in the homework. Please note: 1.) The
homework is to be prepared in groups of two. 2.) Copying from another group will result in zero
points. 3.) All calculations for arriving at an answer must be shown. 4.) If you fail in the exam without
extra points, they cannot be taken into account. 5.) The extra points expire for the resit.

Number of passed worksheets: Extra points for the written exam: (based on 100 achievable points) 0
- 3.5 0 4 - 4.5 4 5 - 5.5 5 6 - 6.5 6

Literature

A. Papoulis, S. Pillai: Probability, Random Variables and Stochastic Processes; McGraw-Hill, 2002
(English) D. Manolakis, V. Ingle, S. Kogon: Statistical and Adaptive Signal Processing; Artech House,
2005 (English)

Timetable: The timetable can be accessed via the StudOn calendar.

Lernziele und Kompetenzen: The
students:

- analyze the statistical properties of random variables, random vectors, and stochastic processes by probability density functions and expectations as well as correlation functions and matrices and their frequency-domain representations
- know the Gaussian distribution and its role to describe the properties of random variables, vectors and processes
- understand the differences between classical and Bayesian estimation, derive and analyze MMSE and ML estimators for specific estimation problems, especially for signal estimation
- analyze and evaluate optimum linear MMSE estimators (single- and multichannel Wiener filter and Kalman filter) for direct and inverse supervised estimation problems
- evaluate adaptive filters for the identification of optimum linear estimators. Die Studenten
- analysieren die statistischen Eigenschaften von Zufallsvariablen, -vektoren und stochastischen Prozessen mittels Wahrscheinlichkeitsdichten und Erwartungswerten, bzw. Korrelationsfunktionen, Korrelationsmatrizen und deren Frequenzbereichsdarstellungen
- kennen die spezielle Rolle der Gaußverteilung und ihre Auswirkungen auf die Eigenschaften von Zufallsvariablen, -vektoren und Prozessen
- verstehen die Unterschiede klassischer und Bayes'scher Schätzung, entwerfen und analysieren MMSE- und ML-Schätzer für spezielle Schätzprobleme, insbesondere zur Signalschätzung
- analysieren und evaluieren lineare MMSE-optimale Schätzer (ein- und vielkanalige Wiener-Filter und Kalman-Filter) für direkte und inverse überwachte Schätzprobleme;
- evaluieren adaptive Filter zur Identifikation optimaler linearer Signalschätzer Literatur:

- A. Papoulis, S. Pillai: Probability, Random Variables and Stochastic Processes; McGraw-Hill, 2002 (englisch)
 - D. Manolakis, V. Ingle, S. Kogon: Statistical and Adaptive Signal Processing; McGraw-Hill, 2005 (englisch)
-

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

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

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern " Communications Engineering (Master of Science)", "123#67#H", "Berufspädagogik Technik (Bachelor of Science)", "Berufspädagogik Technik (Master of Education)", "Communications and Multimedia Engineering (Master of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Elektrotechnik, Elektronik und Informationstechnik (Bachelor of Science)", "Elektrotechnik, Elektronik und Informationstechnik (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Wirtschaftsingenieurwesen (Master of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Statistical Signal Processing (Prüfungsnummer: 64301)

(englische Bezeichnung: Statistical Signal Processing)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90 Anteil

an der Berechnung der Modulnote: 100% weitere

Erläuterungen:

Durch Abgabe der Übungsblätter können Bonuspunkte für die Klausur erarbeitet werden. Wird die Klausur ohne Bonus nicht bestanden, darf der Bonus nicht angerechnet werden. Der Bonus verfällt dann auch für die Wiederholungsklausur. Es gilt folgende Abbildung (bei 100 erreichbaren Punkten in der Klausur): weniger als 4 Übungspunkte = 0 Bonuspunkte in der Klausur, 4 bis 4,5 Übungspunkte = 4 Bonuspunkte in der Klausur, 5 bis 5,5 Übungspunkte = 5 Bonuspunkte in der Klausur, 6 bis 6,5 Übungspunkte = 6 Bonuspunkte in der Klausur, 7 Übungspunkte = 7 Bonuspunkte in der Klausur.

Erstablingung: WS 2018/2019, 1. Wdh.: SS 2019

1. Prüfer: Walter Kellermann

Modulbezeichnung:	Digital Communications (DiCo) (Digital Communications)	5 ECTS
Modulverantwortliche/r:	Laura Cottatellucci	
Lehrende:	Laura Cottatellucci	
Startsemester: WS 2018/2019	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Englisch
Lehrveranstaltungen:		
Digital Communications (WS 2018/2019, Vorlesung, 3 SWS, Laura Cottatellucci et al.)		
Tutorial for Digital Communications (WS 2018/2019, Übung, 1 SWS, Katharina Ackermann et al.)		

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

Nachrichtentechnische Systeme

Inhalt:

Alle modernen Kommunikationssysteme basieren auf digitalen Übertragungsverfahren.

Diese Vorlesung befasst sich mit den Grundlagen der Analyse und des Entwurfs digitaler Sender und Empfänger. Dabei wird zunächst von einem einfachen Kanalmodell bei dem das Empfangssignal nur durch additives weißes Gaußsches Rauschen gestört wird ausgegangen. Im Verlauf der Vorlesung werden aber auch Kanäle mit unbekannter Phase sowie verzerrende Kanäle betrachtet. Behandelt werden unter anderem digitale Modulationsverfahren (z.B. Pulsamplitudenmodulation (PAM), digitale Frequenzmodulation (FSK), und Kontinuierliche-Phasenmodulation (CPM)), Orthogonalkonstellationen, das Nyquistkriterium in Zeit- und Frequenzbereich, optimale kohärente und inkohärente Detektions- und Decodierungsverfahren, die Signalraumdarstellung digital modulierter Signale, verschiedene Entzerrungsverfahren, und Mehrträger-Übertragungsverfahren.

—
Modern communication systems are based on digital transmission methods.

This course covers basics of analysis and design of digital transmitters and receivers. Initially, we consider a simple channel model whose received signal is impaired only by additive white Gaussian noise. Then, we extend fundamental concepts to channels with unknown phases and distortion. Additionally, we treat digital modulation techniques, e.g., pulse amplitude modulation (PAM), digital frequency modulation (FSK) and continuous-phase modulation (CPM), and orthogonal constellations. The Nyquist criterion in time and frequency domain, optimal coherent and incoherent detection and decoding methods, signal space representations of digitally modulated signals, various equalization methods, and multicarrier transmission methods are also discussed.

Lernziele und Kompetenzen:

Die Studierenden

- analysieren und klassifizieren digitale Modulationsverfahren hinsichtlich ihrer Leistungs- und Bandbreiteneffizienz sowie ihres Spitzenwertfaktors,
- ermitteln notwendige Kriterien für impulsinterferenzfreie Übertragung,
- charakterisieren digitale Modulationsverfahren im Signalraum,
- ermitteln informationsverlustfreie Demodulationsverfahren,
- entwerfen optimale kohärente und inkohärente Detektions- und Decodierungsverfahren,
- vergleichen verschiedene Entzerrungsverfahren hinsichtlich deren Leistungsfähigkeit und Komplexität,
- entwerfen einfache digitale Übertragungssysteme mit vorgeschriebenen Leistungs- und Bandbreiteneffizienzen sowie Spitzenwertfaktoren.

—
The students

- analyze and classify digital modulation techniques in terms of performance and bandwidth efficiency as well as crest factor,
- determine necessary criteria to design impulses for interference-free transmission,
- characterize digital modulation methods in signal space,
- determine information loss-free demodulation methods,
- design optimal coherent and incoherent detection and decoding methods,
- compare different equalization methods in terms of performance and complexity,
- design simple digital transmission systems with prescribed power and bandwidth efficiency and crest factor.

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

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

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "123#67#H", "Communications and Multimedia Engineering (Master of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Medizintechnik (Bachelor of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Digital Communications (Prüfungsnummer: 78001)

(englische Bezeichnung: Digital Communications)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90 Anteil
an der Berechnung der Modulnote: 100%

Erstablingung: WS 2018/2019, 1. Wdh.: SS 2019

1. Prüfer: Laura Cottatellucci, 2. Prüfer: Robert Schober

Modulbezeichnung:	Kanalcodierung (KaCo) (Channel Coding)	5 ECTS
Modulverantwortliche/r:	Clemens Stierstorfer	
Lehrende:	Clemens Stierstorfer	
Startsemester: WS 2018/2019	Dauer: 1 Semester	Turnus: halbjährlich (WS+SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch oder Englisch
Lehrveranstaltungen:		
Übungen zur Kanalcodierung (WS 2018/2019, Übung, 1 SWS, Clemens Stierstorfer) Kanalcodierung (WS 2018/2019, 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 umfassende 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-SolomonCodes gemäß der BCH-Schranke. Sie verstehen die Grundlagen der Decodierung von BCH- und ReedSolomon-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] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern " Communications Engineering (Master of Science)", "123#67#H", "Berufspädagogik Technik (Bachelor of Science)", "Berufspädagogik Technik (Master of Education)", "Communications and Multimedia Engineering (Master of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Elektrotechnik, Elektronik und Informationstechnik (Bachelor of Science)", "Elektrotechnik, Elektronik und Informationstechnik (Master of Science)", "Informations- und Kommunikationstechnik (Bachelor of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Mathematik (Bachelor 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: Deutsch oder Englisch

Erstablingung: WS 2018/2019, 1. Wdh.: SS 2019, 2. Wdh.: WS 2019/2020 1.
Prüfer: Clemens Stierstorfer

Modulbezeichnung: Computergraphik-VU (CG-VU) 5 ECTS
(Computer Graphics)

Modulverantwortliche/r: Marc Stamminger

Lehrende: Marc Stamminger

Startsemester: WS 2018/2019 Dauer: 1 Semester Turnus: jährlich (WS)

Präsenzzeit: 60 Std. Eigenstudium: 90 Std. Sprache: Deutsch

Lehrveranstaltungen:

Computergraphik (WS 2018/2019, Vorlesung, 3 SWS, Marc Stamminger) Übungen
zur Computergraphik (WS 2018/2019, Übung, 1 SWS, N.N.)

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

Inhalt:

Die Vorlesung gibt eine Einführung in die Computergraphik:

- Graphik Pipeline
- Clipping
- 3D Transformationen
- Hierarchische Display Strukturen
- Perspektive und Projektionen
- Visibilitätsbetrachtungen
- Rastergraphik und Scankonvertierung
- Farbmodelle
- Lokale und globale Beleuchtungsmodelle
- Schattierungsverfahren
- Ray Tracing und Radiosity
- Schatten und Texturen

Contents:

This lecture covers the following aspects of Computer Graphics:

- graphics pipeline
- clipping
- 3D transformations
- hierarchical display structures
- perspective transformations and projections
- visibility determination
- raster graphics and scan conversion

- color models
- local and global illumination models
- shading models
- ray tracing and radiosity
- shadows and textures

Lernziele und Kompetenzen: Die Studierenden

- geben die unterschiedlichen Schritte der Graphik Pipeline wieder
- erklären die Funktionsweise der Clippingalgorithmen für Linien und Polygone
- beschreiben, charakterisieren und berechnen affine und perspektivische Transformationen in 3D und veranschaulichen die allgemeine Form der Transformationsmatrix in homogener Koordinaten
- skizzieren die Verfahren zur Tiefe- und Visibilitätsberechnung
- vergleichen die unterschiedlichen Farbmodelle der Computergraphik
- illustrieren und untersuchen die Datenstrukturen zur Beschreibung virtueller 3D Modelle und komplexer Szenen
- erläutern die Funktionsweise der Rasterisierung und Scankonvertierung in der Graphikpipeline
- lösen Aufgaben zu Beleuchtung und Texturierung von 3D virtuellen Modellen
- klassifizieren Schattierungsverfahren
- bestimmen den Unterschied zwischen lokaler und globaler Beleuchtung und formulieren Algorithmen für Ray Tracing und Radiosity

Educational objectives and skills:

Students should be able to

- describe the processing steps in the graphics pipeline
 - explain clipping algorithms for lines and polygons
 - explain, characterize and compute affine and perspective transformations in 2D and 3D, and provide an intuitive description of the general form of corresponding transformation matrices in homogeneous coordinates
 - depict techniques to compute depth, occlusion and visibility
 - compare the different color models
 - describe data structures to represent 3D virtual models and complex scenes
 - explain the algorithms for rasterization and scan conversion
 - solve problems with shading and texturing of 3D virtual models
 - classify different shadowing techniques
 - explain the difference between local and global illumination techniques and formulate algorithms for ray tracing and radiosity
- Literatur:
- P. Shirley: Fundamentals of Computer Graphics. AK Peters Ltd., 2002
 - Hearn, M. P. Baker: Computer Graphics with OpenGL. Pearson
 - Foley, van Dam, Feiner, Hughes: Computer Graphics - Principles and Practice
 - Rauber: Algorithmen der Computergraphik
 - Bungartz, Griebel, Zenger: Einführung in die Computergraphik
 - Encarnação, Strasser, Klein: Computer Graphics

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

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

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "123#67#H", "Communications and Multimedia Engineering (Master of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of

Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Informatik (Bachelor of Arts (2 Fächer))", "Informatik (Bachelor of Science)", "Informatik (Master of Science)", "Informations- und Kommunikationstechnik (Bachelor of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Maschinenbau (Bachelor of Science)", "Maschinenbau (Master of Science)", "Mathematik (Bachelor of Science)", "Medizintechnik (Bachelor of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Computergraphik (Vorlesung mit Übung) (Prüfungsnummer: 38201)

Prüfungsleistung, Klausur, Dauer (in Minuten): 60 Anteil

an der Berechnung der Modulnote: 100% weitere

Erläuterungen:

Übung: 50% der schriftlichen Aufgaben, Modulnote durch Klausur über 60 Minuten

Erstablingung: WS 2018/2019, 1. Wdh.: SS 2019

1. Prüfer: Marc Stamminger

Bemerkungen:

Vorlesungsunterlagen, Übungsblätter und die Klausur sind in englischer Sprache

Modulbezeichnung:	Cyber-Physical Systems (CPS) (Cyber-Physical Systems)	5 ECTS
Modulverantwortliche/r:	Torsten Klie	
Lehrende:	Torsten Klie	

Startsemester: WS 2018/2019	Dauer: 1 Semester	Turnus: halbjährlich (WS+SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

Lehrveranstaltungen:

Cyber-Physical Systems (WS 2018/2019, Vorlesung, 2 SWS, Torsten Klie) Übung zu
 Cyber-Physical Systems (WS 2018/2019, Übung, Torsten Klie)

Inhalt:

Klassische Computersysteme zeichnen sich durch eine strikte Trennung von realer und virtueller Welt aus. Moderne Steuerungssysteme, die z.B. in modernen Fahrzeugen verbaut sind und die aus einer Vielzahl von Sensoren und Aktoren bestehen, entsprechen diesem Bild nur sehr eingeschränkt. Diese Systeme, oft "Cyber-physical Systems (CPS)" genannt, erkennen ihre physische Umgebung, verarbeiten diese Informationen und können die physische Umwelt auch koordiniert beeinflussen. Hierzu ist eine starke Kopplung von physischem Anwendungsmodell und dem Computer-Steuerungsmodell nötig. Im Unterschied zu Eingebetteten Systemen bestehen CPS meist aus vielen vernetzten Komponenten, die sich selbständig untereinander koordinieren. Lernziele und Kompetenzen:

- Was sind Cyber-physical Systems? (Definitionen, Abgrenzung zu eingebetteten Systemen, Ubiquitous Computing, etc.)
- Kontrolltheorie und Echtzeitanforderungen
- Selbstorganisationsprinzipien ("Self-X", Autonomie, Verhandlungen)
- Anwendungen für Cyber-physical Systems (Beispiele für existierende oder visionäre zukünftige Anwendungen im Bereich Verkehr, Medizintechnik, u.a.)
- Entwurfsmethoden für Cyber-physical Systems (Modellierung, Programmierung, Model-Integrated Development) Literatur:
- Otto Föllinger Regelungstechnik. Hüthig 1992.
- Rolf Isermann Mechatronische Systeme. Springer 2008.
- Hilmar Jaschek und Holger Voos Grundkurs der Regelungstechnik. Oldenbourg 2010.
- Wolfgang Schneider Praktische Regelungstechnik. Vieweg+Teubner 2008.
- Wayne Wolf Computers as Components: Principles of Embedded Computing System Design. Elsevier 2008

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

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

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "123#67#H", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Informatik (Bachelor of Arts (2 Fächer))", "Informatik (Bachelor of Science)", "Informatik (Master of Science)", "Mathematik (Bachelor of Science)", "Mechatronik (Bachelor of Science)", "Mechatronik (Master of Science)", "Medizintechnik (Bachelor of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Mündliche Prüfung Cyber-Physical Systems (Prüfungsnummer: 636348)

(englische Bezeichnung: Oral Examination on Cyber-Physical Systems)

Prüfungsleistung, Klausur, Dauer (in Minuten): 30
Anteil an der Berechnung der Modulnote: 0%
weitere Erläuterungen:
Die Note ergibt sich zu 100% aus der 90 minütigen Klausur.

Erstablingung: WS 2018/2019, 1. Wdh.: SS 2019
1. Prüfer: Torsten Klie

Cyber-Physical Systems (Prüfungsnummer: 44701)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30
Anteil an der Berechnung der Modulnote: 100% weitere
Erläuterungen:
Die Note ergibt sich zu 100% aus der 30 mündlichen Prüfung.

Erstablingung: WS 2018/2019, 1. Wdh.: SS 2019
1. Prüfer: Torsten Klie

Modulbezeichnung:	Domain-Specific and Resource-Aware Computing on Multicore Architectures (DSC) (Domain-Specific and Resource-Aware Computing on Multicore Architectures)	5 ECTS
Modulverantwortliche/r:	Frank Hannig	
Lehrende:	Frank Hannig	
Startsemester: WS 2018/2019	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch
Lehrveranstaltungen:		
Domain-Specific and Resource-Aware Computing on Multicore Architectures (WS 2018/2019, Vorlesung, 2 SWS, Frank Hannig)		
Übung zu Domain-Specific and Resource-Aware Computing on Multicore Architectures (WS 2018/2019, Übung, 2 SWS, Frank Hannig)		

Inhalt:

Der gegenwärtige Trend von Multi-Core-Architekturen mit mehreren Prozessorkernen hin zu Architekturen mit hunderten oder tausenden Prozessoren bietet ein enormes Potential für schnellere, energieeffizientere, kostengünstigere Systeme und vollkommen neue Anwendungen. Auf der anderen Seite ergeben sich aus der steigenden Komplexität und Strukturgrößen im Nanometerbereich erhebliche Herausforderungen, angefangen bei der Technologie, beim Architektorentwurf bis hin zur Programmierung Systeme basierend auf gemeinsamen Speicher oder zentralverwaltete werden in Zukunft nicht mehr skalieren Hier Bedarf es neuer Architektur- und Programmierkonzepte, um die Skalierbarkeit zu gewährleisten, sowie Methoden zur Optimierung der Ressourcenauslastung, des Leistungsverbrauchs, der Performance und der Toleranz von Fehlern Um diese unterschiedlichen Ziele zu erreichen, werden in der Lehrveranstaltung zwei wesentliche Ansätze betrachtet: *Ressourcenverwaltung / Ressourcengewahre Programmierung* und *Domänenspezifisches Rechnen*. Die Grundidee der ressourcengewahren Programmierung besteht darin, parallelen Programmen die Fähigkeit zu verleihen, selbstadaptiv zur Laufzeit in Abhängigkeit des Zustands von Ressourcen, Berechnungen auf diese zu verteilen, und nach paralleler Abarbeitung diese wieder frei zu geben. Beim domänenspezifischen Rechnen versucht man die oben genannten Ziele durch Einschränkung und Spezialisierung auf ein bestimmtes Anwendungsgebiet oder Problemfeld zu erreichen.

Die Lehrveranstaltung gliedert sich im Wesentlichen in folgende Teile:

- Im ersten Teil werden aktuelle parallele Prozessorarchitekturen vorgestellt und nach unterschiedlichen Merkmalen klassifiziert. Außerdem werden gegenwärtige und zukünftige Herausforderungen von Architekturen und deren Programmierung betrachtet.
- Im zweiten Teil der Veranstaltung werden Abbildungsmethoden und Ansätze, wie zum Beispiel *Invasives Rechnen* zur ressourcengewahren Programmierung für Multi- und Many-Core-Architekturen vorgestellt.
- Domänenspezifisches Rechnen wird im dritten Teil der Lehrveranstaltung betrachtet. Hierbei werden grundlegende Entwurfsmuster und Ansätze domänenspezifischer Sprachen erörtert und an konkreten Beispielen vertieft.

Schwerpunkt der Vorlesung ist die Vermittlung von Grundlagen moderner Multi- und Many-Core-Architekturen und Abbildungstechniken auf diese. Des Weiteren werden Programmierkenntnisse in den Sprachen Scala und X10 erlernt.

Lernziele und Kompetenzen:

Fachkompetenz

Analysieren

- Die Studierenden klassifizieren grundlegende Arten der Parallelverarbeitung und paralleler Prozessorarchitekturen nach unterschiedlichen Merkmalen.

- Die Studierenden diskutieren gegenwärtige und zukünftige Herausforderungen von Multi-CoreArchitekturen.

Evaluieren (Beurteilen)

- Die Studierenden evaluieren und vergleichen Konzepte der *ressourcengewahren Programmierung* und des *domänenspezifischen Rechnens*.

Erschaffen

- Die Studierenden konzipieren und entwickeln einfache *domänenspezifische Programmiersprachen*.
-

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

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

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Domain-Specific and Resource-Aware Computing on Multicore Architectures (Vorlesung mit Übung)
(Prüfungsnummer: 293143)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100%

Erstabelleung: WS 2018/2019, 1. Wdh.: SS 2019

1. Prüfer: Frank Hannig

Modulbezeichnung:	Pattern Recognition (PR) (Pattern Recognition)	5 ECTS
Modulverantwortliche/r:	Andreas Maier	
Lehrende:	Elmar Nöth, Sebastian Käßler	

Startsemester: WS 2018/2019	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Englisch

Lehrveranstaltungen:

- Pattern Recognition (WS 2018/2019, Vorlesung, 3 SWS, Elmar Nöth)
- Pattern Recognition Exercises (WS 2018/2019, Übung, 1 SWS, AmirAbbas Davari et al.)

Empfohlene Voraussetzungen:

- Well grounded in probability calculus, linear algebra/matrix calculus
 - The attendance of our bachelor course 'Introduction to Pattern Recognition' is not required but certainly helpful.
 - Gute Kenntnisse in Wahrscheinlichkeitsrechnung und Linearer Algebra/Matrizenrechnung
 - Der Besuch der Bachelor-Vorlesung 'Introduction to Pattern Recognition' ist zwar keine Voraussetzung, aber sicherlich von Vorteil.
-

Inhalt:

Mathematical foundations of machine learning based on the following classification methods:

- Bayesian classifier
- Logistic Regression
- Naive Bayes classifier
- Discriminant Analysis
- norms and norm dependent linear regression
- Rosenblatt's Perceptron
- unconstraint and constraint optimization
- Support Vector Machines (SVM)
- kernel methods
- Expectation Maximization (EM) Algorithm and Gaussian Mixture Models (GMMs)
- Independent Component Analysis (ICA)
- Model Assessment
- AdaBoost

Mathematische Grundlagen der maschinellen Klassifikation am Beispiel folgender Klassifikatoren:

- Bayes-Klassifikator
- Logistische Regression
- Naiver Bayes-Klassifikator
- Diskriminanzanalyse
- Normen und normabhängige Regression
- Rosenblatts Perzeptron
- Optimierung ohne und mit Nebenbedingungen
- Support Vector Maschines (SVM)
- Kernelmethoden
- Expectation Maximization (EM)-Algorithmus und Gaußsche Mischverteilungen (GMMs)
- Analyse durch unabhängige Komponenten
- Modellbewertung
- AdaBoost

Lernziele und Kompetenzen: Die
Studenten

- verstehen die Struktur von Systemen zur maschinellen Klassifikation einfacher Muster
 - erläutern die mathematischen Grundlagen ausgewählter maschineller Klassifikatoren
 - wenden Klassifikatoren zur Lösung konkreter Klassifikationsprobleme an
 - beurteilen unterschiedliche Klassifikatoren in Bezug auf ihre Eignung
 - verstehen in der Programmiersprache Python geschriebene Lösungen von Klassifikationsproblemen und Implementierungen von Klassifikatoren
 - understand the structure of machine learning systems for simple patterns
 - explain the mathematical foundations of selected machine learning techniques
 - apply classification techniques in order to solve given classification tasks
 - evaluate various classifiers with respect to their suitability to solve the given problem
 - understand solutions of classification problems and implementations of classifiers written in the programming language Python
 - Literatur:
 - Richard O. Duda, Peter E. Hart, David G. Stork: Pattern Classification, 2nd edition, John Wiley&Sons, New York, 2001
 - Trevor Hastie, Robert Tibshirani, Jerome Friedman: The Elements of Statistical Learning - Data Mining, Inference, and Prediction, 2nd edition, Springer, New York, 2009
 - Christopher M. Bishop: Pattern Recognition and Machine Learning, Springer, New York, 2006
-

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

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

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern " Communications Engineering (Master of Science)", "123#67#H", "Communications and Multimedia Engineering (Master of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Informatik (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "International Information Systems (IIS) (Master of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Pattern Recognition (Prüfungsnummer: 41301)

(englische Bezeichnung: Pattern Recognition)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100% weitere

Erläuterungen:

30-minütige mündliche Prüfung über den Stoff der Vorlesung und der Tafelübung

30 minute oral exam about the lecture and the corresponding exercises

Prüfungssprache: Deutsch oder Englisch

Erstablesung: WS 2018/2019, 1. Wdh.: SS 2019

1. Prüfer: Elmar Nöth

Modulbezeichnung:	Pattern Recognition Deluxe (PR) (Pattern Recognition Deluxe)	7.5 ECTS
Modulverantwortliche/r:	Andreas Maier	
Lehrende:	Sebastian Käppler, Elmar Nöth	
Startsemester:	WS 2018/2019	Dauer: 1 Semester
		Turnus: jährlich (WS)

Präsenzzeit: 80 Std.

Eigenstudium: 145 Std.

Sprache: Englisch

Lehrveranstaltungen:

Pattern Recognition (WS 2018/2019, Vorlesung, 3 SWS, Elmar Nöth)

Pattern Recognition Exercises (WS 2018/2019, Übung, 1 SWS, AmirAbbas Davari et al.)

Pattern Recognition Programming (WS 2018/2019, Übung, 2 SWS, AmirAbbas Davari et al.)

Empfohlene Voraussetzungen:

- Well grounded in probability calculus, linear algebra/matrix calculus
- A pattern recognition system consists of the following steps: sensor data acquisition, pre-processing, feature extraction, and classification. Our bachelor course 'Introduction to Pattern Recognition' focuses on the first three steps; this course on the final step of the pipeline, i.e. classification/machine learning. Knowledge about feature extraction is not required for studying the mathematical foundations of machine learning, but it is certainly helpful to get a better understanding of the whole picture.
- Gute Kenntnisse in Wahrscheinlichkeitsrechnung und Linearer Algebra/Matrizenrechnung
- Ein Mustererkennungssystem besteht aus den folgenden Verarbeitungsstufen: Aufnahme der Sensordaten, Vorverarbeitung, Merkmalsextraktion und Klassifikation. Unsere Bachelor-Vorlesung 'Introduction to Pattern Recognition' behandelt hauptsächlich die ersten drei Stufen, während diese Veranstaltung sich mit den mathematischen Grundlagen der Klassifikation/des maschinellen Lernens beschäftigt. Wissen über die Merkmalsextraktion ist für das Verständnis der mathematischen Grundlagen der automatischen Klassifikation zwar nicht notwendig, aber es hilft sicherlich, das Gesamtbild besser zu verstehen.

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

Introduction to Pattern Recognition Deluxe Introduction
to Pattern Recognition

Inhalt:

Mathematical foundations of machine learning based on the following classification methods:

- Bayesian classifier
- Logistic Regression
- Naive Bayes classifier
- Discriminant Analysis
- norms and norm dependent linear regression
- Rosenblatt's Perceptron
- unconstraint and constraint optimization
- Support Vector Machines (SVM)
- kernel methods
- Expectation Maximization (EM) Algorithm and Gaussian Mixture Models (GMMs)
- Independent Component Analysis (ICA)
- Model Assessment
- AdaBoost

Mathematische Grundlagen der maschinellen Klassifikation am Beispiel folgender Klassifikatoren:

- Bayes-Klassifikator
- Logistische Regression
- Naiver Bayes-Klassifikator
- Diskriminanzanalyse
- Normen und normabhängige Regression
- Rosenblatts Perzeptron
- Optimierung ohne und mit Nebenbedingungen

- Support Vector Machines (SVM)
- Kernelmethoden
- Expectation Maximization (EM)-Algorithmus und Gaußsche Mischverteilungen (GMMs)
- Analyse durch unabhängige Komponenten
- Modellbewertung
- AdaBoost

Lernziele und Kompetenzen: Die Studierenden

- verstehen die Struktur von Systemen zur maschinellen Klassifikation einfacher Muster
- erläutern die mathematischen Grundlagen ausgewählter maschineller Klassifikatoren
- wenden Klassifikatoren zur Lösung konkreter Klassifikationsprobleme an
- beurteilen unterschiedliche Klassifikatoren in Bezug auf ihre Eignung
- lösen selbständig Klassifikationsprobleme und schreiben eigene Implementierungen von Klassifikatoren in der Programmiersprache Python

Students

- understand the structure of machine learning systems for simple patterns
 - explain the mathematical foundations of selected machine learning techniques
 - apply classification techniques in order to solve given classification tasks
 - evaluate various classifiers with respect to their suitability to solve the given problem
 - solve classification problems on their own and write their own implementations of classifiers in the programming language Python
- Literatur:
- Richard O. Duda, Peter E. Hart, David G. Stock: Pattern Classification, 2nd edition, John Wiley&Sons, New York, 2001
 - Trevor Hastie, Robert Tibshirani, Jerome Friedman: The Elements of Statistical Learning - Data Mining, Inference, and Prediction, 2nd edition, Springer, New York, 2009
 - Christopher M. Bishop: Pattern Recognition and Machine Learning, Springer, New York, 2006
-

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

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

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "Communications and Multimedia Engineering (Master of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Informatik (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Pattern Recognition Deluxe (Prüfungsnummer: 456863)

(englische Bezeichnung: Pattern Recognition Deluxe)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100% weitere

Erläuterungen:

30-minütige mündliche Prüfung über den Stoff der Vorlesung und der Übungen. Voraussetzung ist die erfolgreiche Bearbeitung der Programmieraufgaben.

30 minute oral exam about the lecture and the exercises. It is required to successfully complete the programming tasks.

Prüfungssprache: Deutsch oder Englisch

Erstablingung: WS 2018/2019, 1. Wdh.: SS 2019

1. Prüfer: Elmar Nöth

Modulbezeichnung:	Multidimensional Signals and Systems (MDSS) (Multidimensional Signals and Systems)	5 ECTS
Modulverantwortliche/r:	Rudolf Rabenstein	
Lehrende:	Rudolf Rabenstein	
Startsemester: WS 2018/2019	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Englisch
Lehrveranstaltungen: Multidimensional Signals and Systems (WS 2018/2019, Vorlesung, 4 SWS, Rudolf Rabenstein)		

Inhalt:

- Typen und Eigenschaften mehrdimensionaler Signale
- Faltung, Delta-Impulse, Abtastung und Fourier-Transformation für mehrdimensionale Signale, Anwendung: Bildverarbeitung
- 2D FIR und IIR Systeme, 2D Zustandsraumdarstellung, Anwendung: Matrixinversion durch Iteration
- Wellenausbreitung in Orts-, Zeit-, Frequenz- und Wellenzahlbereich, Greensche Funktion,

Anwendung: Schallfeldwiedergabe

Contents

Properties of multidimensional signals

- separability, symmetry, etc.

2D signals and systems

- convolution
- delta impulse
- Fourier transformation
- FIR and IIR systems
- state space representation

Wave propagation in 2D and 3D

- wave equation,
- Fourier transformation and decomposition into plane waves and circular and spherical harmonics
- Green's function
- Kirchhoff-Helmholtz integral equation Applications
- imaging with the pin hole camera model
- principle of computer tomography
- subpixel rendering
- iterative solution of systems of linear equations
- room acoustics

sound field reproduction with wave field synthesis and Ambisonics Lernziele und Kompetenzen:

Die Studierenden

- klassifizieren mehrdimensionale Signale und Systeme,
- bewerten mehrdimensionale Abtastverfahren im Orts- und im Wellenzahlbereich
- beurteilen die Eigenschaften mehrdimensionaler FIR und IIR Systeme,
- diskutieren die Vor- und Nachteile von Wellendarstellungen in verschiedenen Bereichen und Koordinatensystemen
- lösen die akustische Wellengleichung in 2 und 3 Ortsdimensionen,
- entwerfen Systeme zur Schallfeldwiedergabe (Ambisonics)
- bewerten die Implementierung von Systemen zur Schallfeldwiedergabe (Wellenfeldsynthese). The students
- classify multidimensional signals and systems,
- assess multidimensional sampling schemes in the space domain and in the wave-number domain,

- evaluate the properties of multidimensional FIR and IIR systems,
 - discuss the advantages and disadvantages of wave representations in different spatial domains and in the corresponding coordinate systems,
 - solve the acoustical wave equation in two and three spatial dimensions,
 - design systems for spatial audio reproduction (Ambisonics)
 - evaluate the implementation of systems for spatial audio reproduction (wave-field synthesis).
-

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

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

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "123#67#H", "Communications and Multimedia Engineering (Master of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Multidimensional Signals and Systems (Prüfungsnummer: 68401)

(englische Bezeichnung: Multidimensional Signals and Systems)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100% Prüfungssprache: Englisch

Erstablingung: WS 2018/2019, 1. Wdh.: SS 2019

1. Prüfer: Rudolf Rabenstein

Organisatorisches:

Mensch-Maschine-Schnittstelle (MMS)

Modulbezeichnung:	Magnetic Resonance Imaging 1 (MRI1) (Magnetic Resonance Imaging 1)	5 ECTS
Modulverantwortliche/r:	Frederik Laun, Andreas Maier, Armin Nagel	
Lehrende:	Armin Nagel, Andreas Maier, Frederik Laun	

Startsemester: WS 2018/2019	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 30 Std.	Eigenstudium: 120 Std.	Sprache: Deutsch oder Englisch

Lehrveranstaltungen:

- Magnetic Resonance Imaging 1 (WS 2018/2019, Vorlesung, 2 SWS, Frederik Laun et al.)
- Magnetic Resonance Imaging 1 - Übung (WS 2018/2019, Übung, Frederik Laun et al.)

Inhalt:

In der Vorlesung werden ausführlich die physikalischen und technischen Grundlagen der MRT behandelt. Es werden der technische Aufbau eines MRTs und die physikalischen Grundlagen behandelt. Das Prinzip der Datenaufnahme wird anhand verschiedener Beispiele erläutert. Fehlkodierungen bei der Datenaufnahme führen zu Bildartefakten, die sich nicht in allen Fällen vermeiden lassen. Strategien zur Erkennung und Vermeidung von Bildartefakten werden erläutert. Eine große Stärke der MRT in der medizinischen Diagnostik ist die Möglichkeit Bilder mit verschiedenen Kontrasten und funktionelle Gewebeparameter aufzunehmen. Die Entstehung der häufig verwendeten T1 und T2 gewichteten Bildkontraste wird ausführlich diskutiert. Des Weiteren werden verschiedene MRT-Sequenztechniken besprochen.

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

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

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "Physics (Master of Science)", "Physik (Bachelor of Science)", "Physik mit integriertem Doktorandenkolleg (Bachelor of Science)", "Physik mit integriertem Doktorandenkolleg (Master of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Magnetic Resonance Imaging 1 (Prüfungsnummer: 122337)

(englische Bezeichnung: Magnetic Resonance Imaging 1)

Prüfungsleistung, Klausur, Dauer (in Minuten): 60 Anteil
an der Berechnung der Modulnote: 100%

Erstablingung: WS 2018/2019, 1. Wdh.: SS 2019

1. Prüfer: Andreas Maier

Modulbezeichnung:	Visual Computing in Medicine (VCMed) (Visual Computing in Medicine)	5 ECTS
Modulverantwortliche/r:	Peter Hastreiter, Thomas Wittenberg	
Lehrende:	Peter Hastreiter, Thomas Wittenberg	

Startsemester: WS 2018/2019	Dauer: 2 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Englisch

Lehrveranstaltungen:

- Visual Computing in Medicine 1 (WS 2018/2019, Vorlesung, 2 SWS, Peter Hastreiter et al.)

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

Algorithmik kontinuierlicher Systeme Computergraphik-
VU

Inhalt:

Die Flut und Komplexität medizinischer Bilddaten sowie die klinischen Anforderungen an Genauigkeit und Effizienz erfordern leistungsfähige wie auch robuste Konzepte der medizinischen Datenverarbeitung. Auf Grund der Vielfalt an Bildinformation und ihrer klinischen Relevanz spielt der Übergang von der Messung medizinischer Bilddaten (u.a. MRT, CT, PET) hin zur Analyse der Bildinhalte eine wichtige Rolle. Durch die visuelle Wiedergabe der abstrakten Daten können sowohl technische als auch medizinische Aspekte anschaulich und intuitiv verstanden werden. Aufbauend auf einem Regelkreis zur Verarbeitung medizinischer Bilddaten werden im ersten Teil (Visual Computing in Medicine I) die Eigenschaften medizinischer Bilddaten sowie grundlegende Methoden und Verfahren der medizinischen Bildanalyse und Visualisierung im Zusammenhang vermittelt. Beispiele aus der Praxis erläutern den Bezug zur medizinischen Anwendung. Darauf aufbauend werden im zweiten Teil (Visual Computing in Medicine II) konkrete Lösungsansätze für die Diagnose und Therapieplanung komplexer Krankheitsbilder erläutert. Es wird gezeigt, wie grundlegende Methoden ausgewählt und zu praktisch anwendbaren Gesamtkonzepten zusammengefasst werden. An Beispielen wird der Bezug zu Strategien und Anforderungen in der industriellen Entwicklung und klinischen Anwendung hergestellt. Ergänzend werden komplexe Methoden der medizinischen Bildanalyse und Visualisierung ausführlich besprochen.

The flood and complexity of medical image data as well as the clinical need for accuracy and efficiency require powerful and robust concepts of medical data processing. Due to the diversity of image information and their clinical relevance the transition from imaging to medical analysis and interpretation plays an important role. The visual representation of abstract data allows understanding both technical and medical aspects in a comprehensive and intuitive way. Based on a processing pipeline for medical image data an overview of the characteristics of medical image data as well as fundamental methods and procedures for medical image analysis and visualization is given. Examples of clinical practice show the relation to the medical application. Based on VCMed1 the lecture VCMed2 discusses practical approaches for the diagnosis and therapy planning of complex diseases. It will be shown how fundamental methods are selected and integrated to practically applicable concepts. Examples demonstrate the relation to strategies and requirements in clinical practice and the industrial development process.

Additionally, complex methods of medical image analysis and visualization will be explained.

Lernziele und Kompetenzen:

Visual Computing in Medicine I

Die Studierenden

- erhalten einen Überblick zu Grundlagen und Unterschieden medizinischer Bildgebungsverfahren
- erwerben fundierte Kenntnisse über Gitterstrukturen, Datentypen und Formate medizinischer Bilddaten
- üben an Beispielen die Erkennung und Interpretation unterschiedlicher Bilddaten
- erwerben Kenntnisse zu Verfahren der Vorverarbeitung, Filterung und Interpolation medizinischer Bilddaten sowie zu grundlegenden Ansätzen der Segmentierung
- erlernen Prinzipien und Methoden der expliziten und impliziten Bildregistrierung und erhalten einen Überblick zu wichtigen Verfahren der starren Registrierung
- erwerben fundierte Kenntnisse zu allen Aspekten der medizinischen Visualisierung (2D, 3D, 4D) von Skalar-, Vektor-, Tensor- und Tensorfeld-Daten
- erhalten an einfachen Beispielen einen ersten Eindruck, wie sich Visualisierung zur Steuerung von

Bildanalyseverfahren und für die medizinische Diagnostik einsetzen lässt The students

- get an overview of the basic principles and differences of medical imaging methods,
- acquire profound knowledge about grid structures, data types and formats of medical image data,
- use sample data to recognize and interpret different image data,
- acquire knowledge about methods of preprocessing, filtering and interpolation of medical image data as well as on basic approaches of segmentation,
- learn the principles and methods of explicit and implicit image registration and get an overview of important procedures of rigid registration,
- acquire profound knowledge about all aspects of medical visualization (2D, 3D, 4D) of scalar, vector, tensor data,
- get a first impression of how visualization can be used to control image analysis and medical diagnostics.

Visual Computing in Medicine II

Die Studierenden

- erwerben aus Sicht der medizinischen Anwendung und konkreter Lösungsstrategien einen Einblick in komplexe Ansätze zur Bearbeitung wichtiger Krankheitsbilder
- lernen die Anforderungen an und die Verknüpfung von Methoden der medizinischen Bildanalyse und Visualisierung zur Bearbeitung kardiologischer, neurologischer, onkologischer und strahlentherapeutischer Fragestellungen
- erhalten einen Überblick zu komplexen Krankheitsbildern als Grundlage für effektive und effiziente Lösungen
- erwerben erweiterte Kenntnisse zur multimodalen Bildregistrierung mit nichtstarrten Transformationen
- erhalten vertieftes Wissen zu komplexen und aktuellen Themen der medizinischen Visualisierung (u.a. Integrationsverfahren, Transferfunktionen, Beschleunigungstechniken mit Grafikkarte)

The students

- gain an insight into complex approaches to the treatment of important disease patterns from the point of view of medical application and specific solution strategies
- learn the requirements and the linking of methods of medical image analysis and visualization for the processing of cardiological, neurological, oncological and radiotherapeutic questions
- get an overview of complex disease pictures as a basis for effective and efficient solutions
- acquire advanced knowledge to process multimodal image data using advanced methods
- receive in-depth knowledge on complex and up-to-date topics of medical visualization (including integration procedures, transfer functions, acceleration techniques with graphics hardware)

Literatur:

- B. Preim, C. Botha: Visual Computing for Medicine, Morgan Kaufmann Verlag, 2013
- B. Preim, D. Bartz: Visualization in Medicine - Theory, Algorithms, and Applications, Morgan Kaufmann Verlag, 2007
- H. Handels: Medizinische Bildverarbeitung, Bildanalyse, Mustererkennung und Visualisierung für die computergestützte ärztliche Diagnostik und Therapie, Vieweg und Teubner Verlag, 2009
- P.M. Schlag, S. Eulenstein, Th. Lange: Computerassistierte Chirurgie, Elsevier Verlag, 2010
- E. Neri, D. Caramella, C. Bartolozzi: Image Processing in Radiology, Springer Verlag, 2008

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

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

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Bachelor of Arts (2 Fächer))", "Informatik (Bachelor of Science)", "Informatik (Master of Science)", "Mathematik (Bachelor of Science)", "Medizintechnik (Bachelor of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Visual Computing in Medicine (Prüfungsnummer: 44811)

(englische Bezeichnung: Visual Computing in Medicine)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100%

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Thomas Wittenberg

Modulbezeichnung:	Biomedizinische Signalanalyse (BioSig) (Biomedical Signal Analysis)	5 ECTS
Modulverantwortliche/r:	Björn Eskofier	
Lehrende:	Björn Eskofier, Heike Leutheuser	

Startsemester: WS 2018/2019	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Englisch

Lehrveranstaltungen:

- Biomedizinische Signalanalyse (WS 2018/2019, Vorlesung, 2 SWS, Björn Eskofier)
- Biomedizinische Signalanalyse Übung (WS 2018/2019, Übung, 2 SWS, Stefan Gradl)

Inhalt:

Im Rahmen der Vorlesung werden (a) die Grundlagen der Generation von wichtigen Biosignalen im menschlichen Körper, (b) die Messung von Biosignalen und (c) Methoden zur Analyse von Biosignalen erläutert und dargestellt.

Behandelte Biosignale sind unter anderem Aktionspotential (AP), Elektrokardiogramm (EKG), Elektromyogramm (EMG), Elektroenzephalogramm (EEG), oder Mechanomyogramm (MMG). Bei der Messung liegt der Fokus beispielsweise auf der Messtechnik oder der korrekten Sensor- bzw. Elektrodenanbringung. Im größten Teil der Vorlesung, Analyse von Biosignalen, werden Konzepte zur Filterung für die Artefaktreduktion, der Wavelet Analyse, der Ereigniserkennung und der Wellenformanalyse behandelt. Zum Schluss wird einen Einblick in überwachte Verfahren der Mustererkennung gegeben. The lecture content explains and outlines (a) basics for the generation of important biosignals of the human body, (b) measurement of biosignals, and (c) methods for biosignals analysis.

Considered biosignals are among others action potential (AP), electrocardiogram (ECG), electromyogram (EMG), electroencephalogram (EEG), or mechanomyogram (MMG). The focus during the measurement part is for example the measurement technology or the correct sensor and electrode placement. The main part of the lecture is the analysis part. In this part, concepts like filtering for artifact reduction, wavelet analysis, event detection or waveform analysis are covered. In the end, an insight into pattern recognition methods is gained.

Lernziele und Kompetenzen:

Die Studierenden

- geben die Entstehung und Messung der wichtigsten Biosignale des menschlichen Körpers wieder
- erkennen Zusammenhänge zwischen der Entstehung der Biosignale des menschlichen Körper und dem gemessenen Signal
- verstehen die Bedeutung der Biosignalverarbeitung für die Medizintechnik
- analysieren die wesentlichen Ursachen von Artefakten in Biosignalen und zeigen Filteroperationen zur Eliminierung dieser Artefakte auf
- wenden erworbenes Wissen über Methoden selbstständig auf interdisziplinäre Fragestellungen der Medizin und der Ingenieurwissenschaften an
- erwerben Schnittstellenkompetenzen zwischen Ingenieurwissenschaften und Medizin
- erlernen fachbezogene Inhalte klar wiederzugeben und argumentativ zu vertreten
- verstehen die Struktur von Systemen zur maschinellen Klassifikation einfacher Muster
- arbeiten in Gruppen kooperativ und verantwortlich
- implementieren Algorithmen der Biosignalverarbeitung in MATLAB
- lösen Klassifikationsprobleme in MATLAB Students
- reproduce the generation and measurement of important biosignals of the human body
- recognize relations between the generation of biosignals and the measured signal
- understand the importance of biosignal analysis for medical engineering
- analyze and provide solutions to the key causes for artifacts in biosignals

- apply gained knowledge independently to interdisciplinary research questions of medicine and engineering science
 - acquire competences between medicine and engineering science
 - learn how to reproduce and argumentatively present subject-related content
 - understand the structure of systems for automatic classification of simple patterns
 - work cooperatively and act responsibly in groups
 - implement biosignal processing algorithms in MATLAB
 - solve classification problems in MATLAB Literatur:
 - R.M. Rangayyan, Biomedical Signal Analysis: A case-study approach. 1st ed., 2002, New York, NY: John Wiley & Sons.
 - E.N. Bruce, Biomedical Signal Processing and Signal Modeling. 1st ed., 2001, New York, NY: John Wiley & Sons.
-

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

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

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "Elektrotechnik, Elektronik und Informationstechnik (Bachelor of Science)", "Elektrotechnik, Elektronik und Informationstechnik (Master of Science)", "Informatik (Bachelor of Arts (2 Fächer))", "Informatik (Bachelor of Science)", "Informatik (Master of Science)", "Mathematik (Bachelor of Science)", "Medizintechnik (Bachelor of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Biomedizinische Signalanalyse (Prüfungsnummer: 30701)

(englische Bezeichnung: Biomedical Signal Analysis)

Prüfungsleistung, Klausur

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

90-minütige schriftliche Klausur über den Stoff der Vorlesung und der Übungen

Erstablingung: WS 2018/2019, 1. Wdh.: SS 2019

1. Prüfer: Björn Eskofier

Modulbezeichnung: Diagnostic Medical Image Processing 5 ECTS
 (VHB-Kurs) (DMIP-VHB)
 (Diagnostic Medical Image Processing (VHB course))

Modulverantwortliche/r: Andreas Maier

Lehrende: Andreas Maier

Startsemester: WS 2018/2019 Dauer: 1 Semester Turnus: halbjährlich (WS+SS)

Präsenzzeit: k.A. Std. Eigenstudium: 150 Std. Sprache: Englisch

Lehrveranstaltungen:

Medical Image Processing for Diagnostic Applications (VHB-Kurs) (WS 2018/2019, Vorlesung, 4 SWS, Andreas Maier)

Empfohlene Voraussetzungen: Ingenieurmathematik

Inhalt:

English version:

The contents of the lecture comprise basics about medical imaging modalities and acquisition hardware. Furthermore, details on acquisition-dependent preprocessing are covered for image intensifiers, flatpanel detectors, and MR. The fundamentals of 3D reconstruction from parallel-beam to cone-beam reconstruction are also covered. In the last chapter, rigid registration for image fusion is explained. In the exercises, algorithms that were presented in the lecture are implemented in Java.

Deutsche Version:

Die Inhalte der Vorlesung umfassen Grundlagen der medizinischen Bildverarbeitung und Aufnahmeprinzipien. Darüber hinaus werden Details der Vorverarbeitung für Bildverstärker, Flachpaneldetektoren und MR erklärt. Die Grundlagen der Rekonstruktion von Parallelstrahl bis hin zur Kegelstrahl-Tomographie werden ebenfalls behandelt. Im letzten Kapitel wird starre Registrierung für Bildfusion erläutert. In den Übungen werden Algorithmen aus der Vorlesung in Java implementiert.

Lernziele und Kompetenzen:

English Version: The participants

- understand the challenges in interdisciplinary work between engineers and medical practitioners.
- develop understanding of algorithms and math for diagnostic medical image processing.
- learn that creative adaptation of known algorithms to new problems is key for their future career.
- develop the ability to adapt algorithms to different problems.
- are able to explain algorithms and concepts of the lecture to other engineers. Deutsche Version:

Die Teilnehmer

- verstehen die Herausforderungen in der interdisziplinären Arbeit zwischen Ingenieuren und Ärzten.
 - entwickeln Verständnis für Algorithmen und Mathematik der diagnostischen medizinischen Bildverarbeitung.
 - erfahren, dass kreative Adaption von bekannten Algorithmen auf neue Probleme der Schlüssel für ihre berufliche Zukunft ist.
 - entwickeln die Fähigkeit Algorithmen auf verschiedene Probleme anzupassen.
 - sind in der Lage, Algorithmen und Konzepte der Vorlesung anderen Studenten der Technischen Fakultät zu erklären.
-

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

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

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "123#67#H", "Communications and Multimedia Engineering (Master of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Informatik (Bachelor of Arts (2 Fächer))", "Informatik (Bachelor of Science)", "Informatik (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Mathematik (Bachelor of Science)", "Medizintechnik (Bachelor of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Diagnostic Medical Image Processing (Prüfungsnummer: 41501)

(englische Bezeichnung: Diagnostic Medical Image Processing)

Prüfungsleistung, Klausur, Dauer (in Minuten): 60

Anteil an der Berechnung der Modulnote: 100% Prüfungssprache: Englisch

Erstablingung: WS 2018/2019, 1. Wdh.: SS 2019

1. Prüfer: Andreas Maier

Modulbezeichnung: Test- und Analyseverfahren zur Softwareverifikation und
-Validierung (TestAn-SWE) 5 ECTS
(Test and Analysis Techniques for Software Ver
Validation)

Modulverantwortliche/r: Francesca Saglietti

Lehrende: Francesca Saglietti

Startsemester: WS 2018/2019 Dauer: 1 Semester Turnus: jährlich (WS)

Präsenzzeit: 60 Std. Eigenstudium: 90 Std. Sprache: Englisch

Lehrveranstaltungen:

Software Test and Analysis (Software Verification and Validation) (WS 2018/2019, Vorlesung, 2 SWS, Francesca Saglietti)

Übungen zu Software Test and Analysis (Exercises in Software Verification and Validation) (WS 2018/2019, Übung, 2 SWS, N.N.)

Inhalt:

Das Modul befasst sich zunächst mit der Bewertung der Relevanz eingebetteter Software in komplexen Automatisierungssystemen. In Abhängigkeit vom Grad der zu übernehmenden Sicherheitsverantwortung werden anschließend zahlreiche Test- und Analyseverfahren unterschiedlicher Rigorosität behandelt, die sich jeweils zur Überprüfung der Entwicklungskorrektheit (Verifikation) bzw. der Aufgabenangemessenheit (Validierung) eignen.

Content:

The module starts with approaches aimed at evaluating the relevance of embedded software in complex control systems. Depending on the degree of the underlying safety relevance, several testing and analysis techniques at different levels of rigour are successively introduced; their application helps checking the correctness of the product developed (verification) resp. the appropriateness of the task specified (validation).

Lernziele und Kompetenzen:

Die Studierenden

- analysieren die Relevanz eingebetteter Software in komplexen Automatisierungssystemen anhand von Fehlerbäumen und kausalen Relationen;
- unterscheiden verschiedene Testverfahren hinsichtlich ihrer Erfüllung struktureller, kontrollflussbasierter bzw. datenflussbasierter Codeüberdeckungskriterien sowie ihres Fehlererkennungspotenzials;
- bewerten die Angemessenheit von Testfallmengen mittels Mutationstesten;
- überprüfen die Korrektheit von Modellen und Programmen anhand axiomatischer Beweisverfahren und Model-Checking-Verfahren.

Learning objectives and competencies:

The students

- analyse the relevance of embedded software in complex control systems by means of fault trees and causal relations;
- distinguish between different testing techniques in terms of their achievement of structural, control flow based resp. data flow based code coverage criteria and their fault detection capabilities;
- evaluate the adequacy of test case sets by means of mutation testing;
- check the correctness of models and programs by means of axiomatic proofs and model checking.

Literatur:

Lehrbuch der Softwaretechnik (Band 1), Helmut Balzert, 2000

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

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

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "123#67#H", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Informatik (Bachelor of Arts (2 Fächer))", "Informatik (Bachelor of Science)", "Informatik (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "International Information Systems (IIS) (Master of Science)", "Mathematik (Bachelor of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Test- und Analyseverfahren zur Software-Verifikation und Validierung (Prüfungsnummer: 32001)

(englische Bezeichnung: Testing and Analytical Procedures in Software Verification and Validation)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100%

Erstablingung: WS 2018/2019, 1. Wdh.: SS 2019

1. Prüfer: Francesca Saglietti

Modulbezeichnung:	Molecular Communications (MolCom) (Molecular Communications)	5 ECTS
Modulverantwortliche/r:	Robert Schober	
Lehrende:	Robert Schober	

Startsemester: WS 2018/2019	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Englisch

Lehrveranstaltungen:

- Molecular Communications (WS 2018/2019, Vorlesung, 4 SWS, Robert Schober)
 - Tutorial for Molecular Communications (WS 2018/2019, Übung, Arman Ahmadzadeh)
-

Inhalt:

Conventional communication systems employ electromagnetic waves for information transmission. This approach is suitable for typical macroscopic applications such as mobile communication. However, newly emerging applications in biology, nanotechnology, and medicine require communication between so-called nano-machines (e.g. nano-robots and nano-sensors) with sizes on the order of nano and micro-meter. For such device sizes electromagnetic waves cannot be used for efficient information transmission. Instead Molecular Communication, an approach that is also widely used in natural biological systems, has to be applied. In Molecular Communication, transmitter and receiver communicate by exchanging information-carrying molecules. The design of molecular communication systems requires a basic understanding of relevant biological processes and systems as well as their communication theoretical modelling and analysis. The course is structured as follows: 1) Introduction to Molecular Communication; 2) Biological Nano-Machines; 3) Molecular Communication in Biological Systems; 4) Synthetic Molecular Communication Systems; 5) Mathematical Modelling and Simulation; 6) Communication and Information Theory for Molecular Communication; 7) Design of Molecular Communication Systems; 8) Applications for Molecular Communication Systems.

Lernziele und Kompetenzen:

The students learn how to design synthetic molecular communication systems. They develop an understanding of natural communication processes in biological systems and how to harness these natural processes for the construction of man-made molecular communication systems. The students also learn how to analyse, model, and simulate molecular communication systems.

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

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

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern " Communications Engineering (Master of Science)", "Communications and Multimedia Engineering (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Molecular Communications (Prüfungsnummer: 454183)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 0% Prüfungssprache: Englisch

Erstablingung: WS 2018/2019, 1. Wdh.: SS 2019

1. Prüfer: Robert Schober

Modulbezeichnung:	eBusiness Technologies und Evolutionäre Informationssysteme (EBTEIS) (eBusiness Technologies and Evolutionary Information Systems)	5 ECTS
Modulverantwortliche/r:	Richard Lenz	
Lehrende:	Richard Lenz, Christoph P. Neumann, Florian Irmert	

Startsemester: WS 2018/2019	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

Lehrveranstaltungen:

Ausschlussbedingung: Dieses Modul darf nur abgelegt werden, wenn keine der im Modul enthaltenen Lehrveranstaltungen auch noch in einem anderen Modul enthalten ist, das bereits abgelegt wurde.

eBusiness Technologies (WS 2018/2019, Vorlesung, Christoph P. Neumann et al.)

Evolutionäre Informationssysteme (WS 2018/2019, Vorlesung, 2 SWS, Richard Lenz)

Empfohlene Voraussetzungen:

Programmieren in Java, Datenbanken (SQL)

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

Inhalt:

siehe Lehrveranstaltungsbeschreibungen

Lernziele und Kompetenzen:

EBT:

Die Studierenden

- identifizieren die wichtigsten Themen des Bereichs eBusiness, von den Anwendungen bis zu den Implementierungen
- verstehen Zusammenhänge der B2B-Integration und der Realisierung von eBusiness-Anwendungen
- wiederholen Grundlagen des Webs
- vergleichen technische Eigenschaften von HTTP-, Web- und Application Servern
- vergleichen Markup Languages (HTML, XML)
- unterscheiden Ansätze zur Schema-Modellierung wie DTD und XML Schema und erkennen die unterschiedliche Leistungsfähigkeit
- verstehen Methoden zur evolutionsfähigen Gestaltung von Datenstrukturen in XML
- unterscheiden Vorgehen bei der Datenhaltung und verschiedene Ansätze für den Datenbankzugriff
- verstehen Objekt-relationale Mapping Frameworks am Beispiel von Hibernate und JPA
- verstehen Komponentenmodelle wie Enterprise JavaBeans (EJB) aus dem JEE Framework
- unterscheiden das EJB Komponentenmodell von den OSGi Bundles und den Spring Beans
- verstehen und unterscheiden grundlegende Web Service Techniken wie SOAP und WSDL
- unterscheiden Herangehensweisen zur dynamischen Generierung von Webseiten
- verstehen grundlegende Eigenschaften eines Java-basierten Front-End-Frameworks am Beispiel von JSF
- verstehen grundlegende Eigenschaften von Service-orientierten Architekturen (SOA)
- verstehen agile Vorgehensmodelle zur Software-Entwicklung am Beispiel von Scrum
- unterscheiden agile Verfahren wie Scrum von iterativ-inkrementellen Verfahren wie RUP
- verstehen die Wichtigkeit von Code-Beispielen um die praktische Anwendbarkeit des theoretischen Wissens zu veranschaulichen.
- können die Code-Beispiele eigenständig zur Ausführung bringen und die praktischen Erfahrungen interpretieren und bewerten
- gestalten eigene Lernprozesse selbständig.

- schätzen ihre eigenen Stärken und Schwächen im Hinblick auf die unterschiedlichen Architekturschichten ein (Benutzerinteraktion, Applikationslogik, Schnittstellenintegration, Datenbanksysteme)
- identifizieren eine eigene Vorstellung als zukünftige Software-Architekten und können die eigene Entwicklung planen
- reflektieren durch regelmäßige fachbezogene Fragen des Dozenten Ihre eigene Lösungskompetenz.

EIS:
Die Studierenden:

- definieren die Begriffe "Informationssysteme", "evolutionäre Informationssysteme" und "organisatorisches Lernen"
 - grenzen die Begriffe "Wissen" und "Information" gegeneinander ab
 - charakterisieren die in der Vorlesung erläuterten Formen der organisatorischen Veränderung
 - erklären das SEKI Modell nach Nonaka und Takeuchi
 - nennen Beispiele für die in der Vorlesung behandelten Formen der Wissensrepräsentation in IT-Systemen
 - nennen typische Erfolgs- und Risikofaktoren für große IT-Projekte
 - erklären die Kraftfeldtheorie nach Kurt Lewin
 - unterscheiden Typen von Software gemäß der Klassifikation nach Lehman und Belady
 - unterscheiden die in der Vorlesung vorgestellten Arten der Software-Wartung
 - benennen die Gesetzmäßigkeiten der Software-Evolution nach Lehman und Belady
 - bewerten die in der Vorlesung vorgestellten Vorgehensmodelle zur Softwareerstellung im Kontext der E-Typ-Software
 - nennen die in der Vorlesung vorgestellten Aspekte der Evolutionsfähigkeit von Software
 - erklären, wie die in der Vorlesung vorgestellten Methoden zur Trennung von Belangen beitragen
 - erklären das Konzept des "Verzögerten Entwurfs"
 - erklären die Vor- und Nachteile generischer Datenbankschemata am Beispiel von EAV und EAV/CR
 - charakterisieren die in der Vorlesung vorgestellten Architekturkonzepte
 - grenzen die in der Vorlesung vorgestellten Integrationsanforderungen gegeneinander ab
 - erklären wie Standards zur Systemintegration beitragen und wo die Grenzen der Standardisierung liegen
 - erklären das Prinzip eines Kommunikationsservers und der nachrichtenbasierten Integration
 - erklären den Begriff "Prozessintegration"
 - definieren den Begriff "Enterprise Application Integration" (EAI)
 - unterscheiden die in der Vorlesung vorgestellten Integrationsansätze
 - erklären die in der Vorlesung vorgestellten Dimensionen der Datenqualität
 - unterscheiden die grundlegenden Messmethoden für Datenqualität
 - erklären das Maßnahmenportfolio zur Verbesserung der Datenqualität nach Redman
 - benennen die in der Vorlesung vorgestellten Methoden zur Verbesserung der Datenqualität
- Literatur: siehe Lehrveranstaltungsbeschreibungen
-

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

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

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Bachelor of Arts (2 Fächer))", "Informatik (Bachelor of Science)", "Informatik (Master of Science)", "International Information Systems (IIS) (Master of Science)", "Mathematik (Bachelor of Science)" verwendbar.

Studien-/Prüfungsleistungen:

eBusiness Technologies und Evolutionäre Informationssysteme (Prüfungsnummer: 710850)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 0%

Erstablingung: WS 2018/2019, 1. Wdh.: SS 2019 (nur für Wiederholer) 1.

Prüfer: Richard Lenz

Modulbezeichnung: Computerunterstützte Messdatenerfassung (CM) 5 ECTS
 (Computer Aided Data Acquisition)

Modulverantwortliche/r: Reinhard Lerch

Lehrende: Reinhard Lerch

Startsemester: WS 2018/2019 Dauer: 1 Semester Turnus: jährlich (WS)

Präsenzzeit: 60 Std. Eigenstudium: 90 Std. Sprache: Deutsch

Lehrveranstaltungen:

Computerunterstützte Messdatenerfassung (WS 2018/2019, Vorlesung, 2 SWS, Reinhard Lerch)

Übungen zu Computerunterstützte Messdatenerfassung (WS 2018/2019, Übung, 2 SWS, Michael Fink)

Inhalt:

Buch: "Elektrische Messtechnik", 7. Aufl. 2016, Springer Verlag, Kap. 11, 13, 15 bis 20

- Analoge Messschaltungen
- Digitale Messschaltungen
- AD-/DA-Wandler
- Messsignalverarbeitung und Rauschen • Korrelationsmesstechnik
- Rechnergestützte Messdatenerfassung
- Bussysteme
- Grundlagen zu Speicherprogrammierbaren Steuerungen

Lernziele und Kompetenzen: Die Studierenden

- kennen die grundlegenden Konzepte und Schaltungen bei der Messung elektrischer Größen
- wählen geeignete Verfahren zur Analyse elektrischer Netzwerke und wenden diese an
- verstehen prinzipielle Methoden der Elektrischen Messtechnik, wie die Korrelationsmesstechnik
- interpretieren Messergebnisse anhand von Methoden der Fehlerrechnung
- kennen Ursachen von Rauschen in elektrischen Netzwerken
- analysieren das Rauschverhalten in elektrischen Netzwerken
- führen Dimensionierungen von Mess- und Auswerteschaltungen durch
- kennen wichtige Hard- und Software-Komponenten zur rechnergestützten Messdatenerfassung
- verstehen Grundprinzipien und Grundschaltungen von AD-/DA-Wandlern
- vergleichen analoge und digitale Verfahren zur Auswertung und Konditionierung von Messsignalen
- kennen und bedienen Messdatenerfassungssysteme für die Laborautomation und die Prozesstechnik

Literatur:

Lerch, R.; Elektrische Messtechnik; 7. Aufl. 2016, Springer Verlag

Lerch, R.; Elektrische Messtechnik - Übungsbuch; 2. Aufl. 2005, Springer Verlag

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

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

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "Berufspädagogik Technik (Bachelor of Science)", "Berufspädagogik Technik (Master of Education)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Elektrotechnik, Elektronik und Informationstechnik (Bachelor of Science)", "Elektrotechnik, Elektronik und Informationstechnik (Master of Science)", "Mechatronik (Bachelor of Science)", "Mechatronik (Master of Science)", "Medizintechnik (Bachelor of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Computerunterstützte Messdatenerfassung_ (Prüfungsnummer: 23401)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90 Anteil

an der Berechnung der Modulnote: 100%

Erstablingung: WS 2018/2019, 1. Wdh.: SS 2019, 2. Wdh.: WS 2019/2020 1.

Prüfer: Reinhard Lerch

Modulbezeichnung:	Lasers in Healthcare Engineering (LASHE) (Lasers in Healthcare Engineering)	2.5 ECTS
Modulverantwortliche/r:	Florian Klämpfl	
Lehrende:	Florian Klämpfl	
Startsemester: WS 2018/2019	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 30 Std.	Eigenstudium: 45 Std.	Sprache: Englisch
Lehrveranstaltungen: Lasers in Healthcare Engineering (WS 2018/2019, Vorlesung, 2 SWS, Florian Klämpfl et al.)		

Empfohlene Voraussetzungen:

The course targets senior Bachelor and Master students who are interested in gaining knowledge about medical applications of lasers. The course combines lecture material (approximately 60%) and practical training (approximately 40%). The course is suitable for students with diverse educational background and no prior knowledge of laser and laser technology is required although general understanding of physics is presumed.

Inhalt:

- Physical phenomena applicable in Laser Technology: EM waves, Beam Propagation;
- Laser tissue interaction processes and Monte-Carlo simulation method;
- Introduction to Optical Coherence Technology;
- Lasers for medical applications;
- Lasers for production of medical tools;
- Optical diagnostic and treatment methods in medicine: laser surgery, Raman spectroscopy, optical phantom preparation and characterization;

Lernziele und Kompetenzen:

Students...

- Would know the fundamentals of laser tissue-interaction process.
- Will understand principles of tissue / phantom optical properties characterization.
- Will be able to perform characterization of basic optical properties of tissues.
- Will gain basic understanding and practical experience with Optical Coherence Tomography (OCT).
 - Will be familiar with potential applications of laser in medicine and healthcare
- Will become familiar with international (English) professional terminology.

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

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

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Studien-/Prüfungsleistungen:

Lasertechnik für die Medizintechnik (Prüfungsnummer: 74601)

(englische Bezeichnung: Medical Laser Technology)

Prüfungsleistung, Klausur

Anteil an der Berechnung der Modulnote: 100%

Erstablingung: WS 2018/2019, 1. Wdh.: SS 2019

1. Prüfer: Michael Schmidt, 2. Prüfer: Florian Klämpfl

Bemerkungen:

Lasertechnik für Master Medizintechnik

Modulbezeichnung:	Einführung in die IT-Sicherheit (EinfITSec) (Introduction to IT Security)	5 ECTS
Modulverantwortliche/r:	Felix Freiling	
Lehrende:	Felix Freiling	
Startsemester: WS 2018/2019	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch
Lehrveranstaltungen:		
Angewandte IT-Sicherheit (WS 2018/2019, Vorlesung, 2 SWS, Felix Freiling et al.)		
Einführung in die IT-Sicherheit - Übung (WS 2018/2019, Übung, 2 SWS, Felix Freiling et al.)		

Inhalt:

Die Vorlesung gibt einen einführenden Überblick über Konzepte und Methoden der IT-Sicherheit und eignet sich als Einstieg in das Vertiefungsgebiet "IT-Sicherheit" an der FAU. Themen (unter anderem): IT-Sicherheit vs. physische Sicherheit, Identifizierung und Authentifizierung, grundlegende Muster von Unsicherheit in technischen Systemen, grundlegende Abwehrmechanismen, ausgewählte Beispiele aus dem Bereich Systemsicherheit, Netzwerksicherheit und Softwaresicherheit. In der Übung werden die Themen der Veranstaltung beispielhaft eingeübt. Themen (unter anderem): "lock picking", "social engineering", ausnutzen von Softwareschwachstellen.

Lernziele und Kompetenzen:

Teilnehmer erwerben einen Überblick über Konzepte und Methoden aus dem Bereich der IT-Sicherheit und können diese im Kontext der Informatik und der Lebenswirklichkeit anhand von Beispielen einordnen und erläutern. Die Studierenden können die wichtigsten Arten von Softwareschwachstellen in Programmen erkennen und benennen. Sie können außerdem erläutern, wie man diese Schwachstellen ausnutzt und welche technischen und organisatorischen Maßnahmen geeignet sind, diese Schwachstellen zu vermeiden. Die Studierenden lernen, die Wirksamkeit von IT-Sicherheitsmechanismen im gesellschaftlichen Kontext und in Kenntnis professioneller Strukturen der Cyberkriminalität aus technischen, ethischen und rechtlichen Perspektiven zu bewerten.

Literatur:

Dieter Gollmann: Computer Security. 3. Auflage, Wiley, 2010. Joachim Biskup: Security in Computing Systems. Springer, 2008. Weitere Literatur wird in der Vorlesung bekanntgegeben.

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

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

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Bachelor of Arts (2 Fächer))", "International Information Systems (IIS) (Master of Science)", "Mathematik (Bachelor of Science)", "Mechatronik (Bachelor of Science)", "Mechatronik (Master of Science)", "Medizintechnik (Bachelor of Science)", "Wirtschaftsinformatik (Bachelor of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Einführung in die IT-Sicherheit (AppITSec-Ü) (Prüfungsnummer: 46311)

(englische Bezeichnung: Applied IT security (AppITSec))

Prüfungsleistung, Klausur

Anteil an der Berechnung der Modulnote: 0%

Erstablingung: WS 2018/2019, 1. Wdh.: SS 2019

1. Prüfer: Felix Freiling

Modulbezeichnung:	Convex Optimization in Communications and Signal Processing (ConvOpt) (Convex Optimization in Communications and Signal Processing)		5 ECTS
Modulverantwortliche/r:	Wolfgang Gerstacker		
Lehrende:	Wolfgang Gerstacker		
Startsemester: WS 2018/2019	Dauer: 1 Semester	Turnus: jährlich (WS)	
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Englisch	

Lehrveranstaltungen:

Convex Optimization in Communications and Signal Processing (WS 2018/2019, Vorlesung, 3 SWS, Wolfgang Gerstacker)

Tutorial for Convex Optimization in Communications and Signal Processing (WS 2018/2019, Übung, 1 SWS, Viktoria Schram)

Empfohlene Voraussetzungen:

Signals and Systems, Communications

Inhalt:

Convex optimization problems are a special class of mathematical problems which arise in a variety of practical applications. In this course we focus on the theory of convex optimization, corresponding algorithms, and applications in communications and signal processing (e.g. statistical estimation, allocation of resources in communications networks, and filter design). Special attention is paid to recognizing and formulating convex optimization problems and their efficient solution. The course is based on the textbook "Convex Optimization" by Boyd and Vandenberghe and includes a tutorial in which many examples and exercises are discussed.

Lernziele und Kompetenzen: Students

- characterize convex sets and functions,
- recognize, describe and classify convex optimization problems,
- determine the solution of convex optimization problems via the dual function and the KKT conditions,
- apply numerical algorithms in order to solve convex optimization problems,
- apply methods of convex optimization to different problems in communications and signal processing

Literatur:

Boyd, Steven ; Vandenberghe, Lieven: Convex Optimization. Cambridge, UK : Cambridge University Press, 2004

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

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

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern " Communications Engineering (Master of Science)", "123#67#H", "Berufspädagogik Technik (Bachelor of Science)", "Berufspädagogik Technik (Master of

Education)", "Communications and Multimedia Engineering (Master of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Elektrotechnik, Elektronik und Informationstechnik (Bachelor of Science)", "Elektrotechnik, Elektronik und Informationstechnik (Master of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Convex Optimization in Communications and Signal Processing (Prüfungsnummer: 68501)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90 Anteil

an der Berechnung der Modulnote: 100%

Erstablingung: WS 2018/2019, 1. Wdh.: SS 2019

1. Prüfer: Wolfgang Gerstacker

Modulbezeichnung: Image, Video, and Multidimensional Signal Processing (IVMSP)
(Image, Video, and Multidimensional Signal Processing) 5 ECTS

Modulverantwortliche/r: André Kaup

Lehrende: André Kaup

Startsemester: WS 2018/2019 Dauer: 1 Semester Turnus: jährlich (WS)

Präsenzzeit: 60 Std. Eigenstudium: 90 Std. Sprache: Englisch

Lehrveranstaltungen:

Image, Video, and Multidimensional Signal Processing (WS 2018/2019, Vorlesung, 3 SWS, André Kaup)
Supplements for Image, Video, and Multidimensional Signal Processing (WS 2018/2019, Übung, 1 SWS, Andreas Spruck)

Empfohlene Voraussetzungen:

Vorlesung Signale und Systeme I und II

Inhalt:

Point operations
Histogram equalization, gamma correction
Binary operations
Morphological filters, erosion, dilation, opening, closing
Color spaces
Trichromacy, red-green-blue color spaces, color representation using hue, saturation and value of intensity
Multidimensional signals and systems
Theory of multidimensional signals and systems, impulse response, linear image filtering, power spectrum, Wiener filtering
Interpolation of image signals
Bi-linear interpolation, bi-cubic interpolation, spline interpolation
Image feature detection
Image features, edge detection, Hough transform, Harris corner detector, texture features, cooccurrence matrix
Scale space representation
Laplacian of Gaussian, difference of Gaussian, scale invariant feature transform, speeded-up robust feature transform
Image matching
Projective transforms, block matching, optical flow, feature-based matching using SIFT and SURF, random sample consensus algorithm
Image segmentation
Amplitude thresholding, k-means clustering, Bayes classification, region-based segmentation, combined segmentation and motion estimation, temporal segmentation of video
Transform domain image processing
Unitary transform, Karhunen-Loeve transform, separable transform, Haar and Hadamard transform, DFT, DCT

Lernziele und Kompetenzen: Die Studierenden

- verstehen Punktoperationen an Bilddaten und Gamma-Korrektur
- testen die Wirkung von Rangordnungs- und Medianfiltern an Bilddaten

- unterscheiden und bewerten verschiedene Farbräume für Bilddaten
- erklären das Prinzip der zwei-dimensionalen linearen Filterung für Bildsignale
- berechnen und bewerten die zweidimensionale diskrete Fourier-Transformierte eines Bildsignales
- bestimmen vergrößerte diskrete Bildsignale mit Methoden der bilinearen und Spline-Interpolation
- überprüfen Bilddaten auf ausgewählte Textur-, Kanten- und Bewegungsmerkmale
- analysieren Bild- und Videodaten auf Merkmale in unterschiedlichen Scale-Spaces
- erläutern und beurteilen Methoden für das Matching von Bilddaten
- segmentieren Bilddaten durch Programmierung von einfachen Klassifikations- oder ClusteringVerfahren
- verstehen das Prinzip von Transformation auf Bilddaten und können diese an Beispielen anwenden.

The students

- understand point operations for image data and gamma correction
- test the effects of rank order and median filters for image data
- evaluate and differentiate between different color spaces for image data
- explain the principle of two-dimensional linear filtering for image signals
- calculate and evaluate the two-dimensional discrete Fourier transform of an image signal
- determine enlarged discrete image signals by bi-linear and spline interpolation
- verify image data for selected texture, edge and motion features
- analyze image and video data for features in different scale spaces
- explain and evaluate methods for the matching of image data
- segment image data by implementing basic classification and clustering methods
- understand the principle of transformations on image data and apply them exemplarily Literatur:

J.-R. Ohm: *Multimedia Content Analysis*, Springer, 2016

J. W. Woods: *Multidimensional Signal, Image, and Video Processing and Coding*, Academic Press, 2nd edition, 2012

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

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

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern " Communications Engineering (Master of Science)", "Communications and Multimedia Engineering (Master of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Image, Video, and Multidimensional Signal Processing (Prüfungsnummer: 447324)

(englische Bezeichnung: Image, Video, and Multidimensional Signal Processing)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 0% Prüfungssprache: Englisch

Erstablingung: WS 2018/2019, 1. Wdh.: SS 2019

1. Prüfer: André Kaup

Bemerkungen:

This lecture replaces the previous lecture Visual Computing for Communication (VCC).