

sss & sssMOR

Analyse und Reduktion dynamischer Systeme sehr hoher Ordnung

A. Castagnotto, M. Cruz Varona, T. Emmert, S. Jaensch, M. Meindl, B. Lohmann, W. Polifke

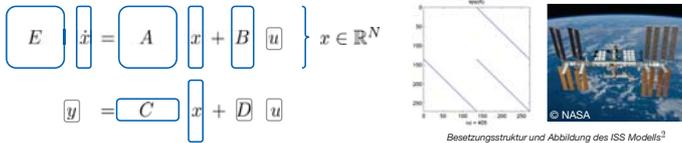
Kurzfassung

Die genaue Modellierung dynamischer Systeme resultiert oft in eine sehr hohe Anzahl an Zustandsvariablen. Die Systemmatrizen können dadurch zu groß werden, um noch Zustandsraummodelle (ss Objekte) in MATLAB¹ definieren zu können. In diesem Beitrag werden Toolboxen vorgestellt, welche durch Einführung sog. **sparse state space Objekten (sss)** die Analyse hochdimensionaler Systeme ermöglichen. Mittels **Modellordnungsreduktion (sssMOR)** kann die wesentliche Dynamik in Modellen deutlich niedrigerer Ordnung abgebildet werden.



Dünnbesetztheit der Systemmatrizen ausnutzen

Lineare zeitinvariante Systeme werden oft zum Zwecke des Reglerentwurfs durch Zustandsraummodelle repräsentiert. Wenn die Modellordnung N sehr hoch ist ($N \gg 10^3$), sind die Systemmatrizen i.A. dünnbesetzt (engl.: *sparse*), sodass die Anzahl an nicht null Einträgen wesentlich kleiner ist als N^2 .



Die **Control System Toolbox** in MATLAB kann leider diese Eigenschaft nicht ausnutzen und speichert alle Matrizen als "full" ab. Aus diesem Grund ist die Definition von Zustandsraummodellen mithilfe der Befehle

```
sys = ss(A,B,C,D) oder sys = dss(A,B,C,D,E)
```

auf einem Standardrechner nur bis zu einer Größenordnung von $\mathcal{O}(10^4)$ möglich. Tatsächlich erfordert die Definition einer vollen Einheitsmatrix der Dimension 10^3 80GB, während die dünnbesetzte Matrix nur noch 2.4MB benötigt!

Funktionalität

Mit **sss** kann die "sparsity" der Matrizen ausgenutzt werden mit erheblichen Vorteilen in puncto Speicher- und Rechenaufwand. Dabei wird ein Zustandsraummodell einfach mit

```
sys = sss(A,B,C,D,E)
```

definiert. Weiterhin enthält die **sss** Toolbox viele Analysefunktionen, die aus der Control System Toolbox bekannt sind, und nutzt gezielt die Vorteile dünnbesetzter Matrizen aus.

Funktionen

Manipulation:

```
>> truncate(sys,p,m); connect(sys1,sys2);...
>> sys1-sys2; sys1*sys2; c2d(sysC,Ts);...
```

Analyse im Frequenzbereich:

```
>> freqresp(sys,w); bode(sys); sigma(sys);...
```

Analyse im Zeitbereich:

```
>> impulse(sys); step(sys); lsim(sys,u,Ts);...
```

Zusätzliche Eigenschaften:

```
>> sys.isDae; sys.isSym; sys.isSimo;...
>> norm(sys,2); norm(sys,inf); isstable(sys);...
>> eigs(sys); spy(sys); diag(sys);...
```

Kompatibilität

Alter Code:

```
>> sys = ss(A,B,C,D)
>> myCode(sys)
```

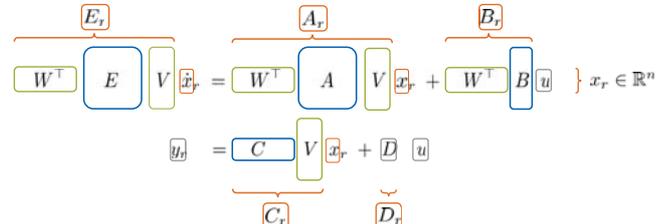
Neuer Code:

```
>> sys = sss(A,B,C,D)
>> myCode(sys)
```



Relevante Dynamik in Modellen niedriger Ordnung erfassen

Auch wenn man die sss Toolbox verwendet, sind Berechnungen mit hochdimensionalen Modellen sehr aufwändig. Aufgrund dessen, werden **reduzierte Modelle** viel niedrigerer Ordnung $n \ll N$ gesucht, die das Übertragungsverhalten gut approximieren. Für lineare Systeme lässt sich die Reduktion als Petrov-Galerkin Projektion der Form

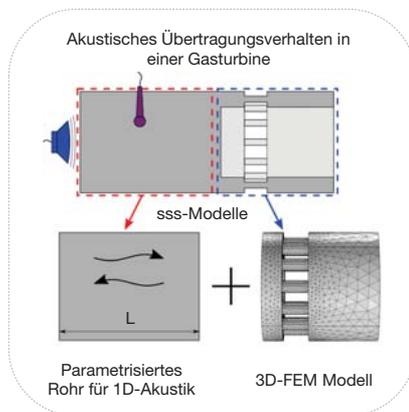


darstellen. Die Projektionsmatrizen V, W können mit unterschiedlichen Verfahren berechnet werden, je nach Eigenschaften vom Originalmodell die erhalten bleiben sollen. Klassische Methoden umfassen *modale Reduktion, Balanciertes Abschneiden und Krylow-Verfahren*, während *IRKA* und *CURED SPARK* Beispiele für fortgeschrittene Algorithmen sind.

Funktionalität

Modellreduktion in **sssMOR** wird durchgeführt, indem der entsprechenden Funktion ein **sss** Objekt des Originalmodells übergeben wird, eventuell mit weiteren Parametern.

Funktionen	Beschreibung
<code>modalMor(sys,n)</code>	Modale Reduktion mit Erhaltung dominanter Eigenwerte
<code>tbr(sys,n)</code>	Balanciertes Abschneiden mit Erhaltung dominanter Hankel Singulärwerte
<code>rk(sys,s0)</code>	Krylow-Unterraum-Methoden mit Matching einiger Taylor Koeffizienten der Übertragungsfunktion
<code>irka(sys,s0)</code>	Iterativer Rationaler Krylow Algorithmus für \mathcal{H}_2 -optimale Reduktion
<code>cirka(sys,s0)</code>	Confined IRKA Algorithmus für schnelle \mathcal{H}_2 -optimale Reduktion
<code>spark(sys,s0)</code>	Stabilitätserhaltender, adaptiver rationaler Krylow Algorithmus
<code>porkV(...)</code>	\mathcal{H}_2 -pseudo-optimaler rationaler Krylow Algorithmus
<code>cure(sys)</code>	Kumulative Reduktion mit adaptiver Wahl der reduzierten Ordnung



Anwendung von sss & sssMOR auf parametrische thermoakustische Netzwerkmodelle in taX

taX ist eine MATLAB Toolbox, mit der man thermoakustische Netzwerkmodelle erzeugen kann. Aufgrund der Dünnbesetztheit der Matrizen werden Zustandsraummodelle als sss Objekte definiert.

Aufbau des Gesamtsystems durch Verbinden der beiden Einzelsysteme:

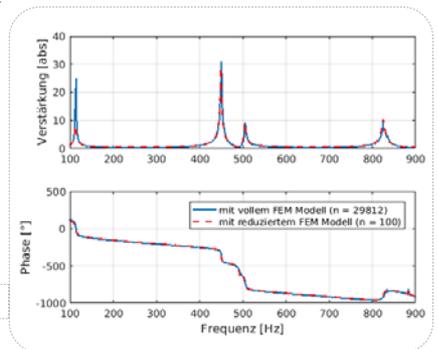
```
>> sysGasturbine = connect(sysRohr, sysFEM)
```

Reduktion des FEM-Modells mit dem **irka** Algorithmus aus **sssMOR**:

```
>> sysFEM_r = irka(sysFEM)
>> sysGasturbine_r = connect(sysRohr, sysFEM_r)
```

Plot des Übertragungsverhaltens mit der **bode** Funktion aus **sss**:

```
>> bode(sysGasturbine, sysGasturbine_r)
```



sss und **sssMOR** sind Open-Source Toolboxen vertrieben unter GPLv2 um den Austausch im Bereich hochdimensionaler Anwendungen sowie Modellreduktion zu fördern. Mehr Informationen unter www.rtf.mw.tum.de/?sss oder www.rtf.mw.tum.de/?sssMOR.

¹MATLAB und die Control System Toolbox (Release 2015b) sind eingetragene Marken von The MathWorks, Inc., Natick, Massachusetts, United States.

²SLICOT Benchmark Modelle: <http://slicot.org/20-site/126-benchmark-examples-for-model-reduction>



Numerische Simulation und analytische Lösung von sub-diffusivem Verhalten bei der Partikel-Diffusion in eingeschränkten Geometrien

Matthias Ernst¹, Thomas John², Marco Günther¹, Christian Wagner², Ulrich F. Schäfer³, Claus-Michael Lehr^{3,4}

¹ Fakultät für Ingenieurwissenschaften, Angewandte Mathematik, Hochschule für Technik und Wirtschaft des Saarlandes, Göbenstraße 40, 66117 Saarbrücken, Germany, matthias.ernst@htwsaar.de

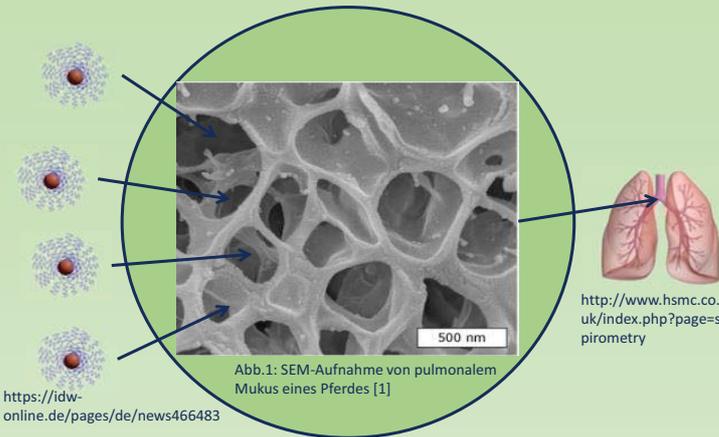


² Experimentalphysik, Universität des Saarlandes, 66123 Saarbrücken, Germany
³ Biopharmazie und Pharmazeutische Technologie, Universität des Saarlandes, 66123 Saarbrücken, Germany
⁴ Helmholtz Institut für Pharmazeutische Forschung Saarland, Universität des Saarlandes, 66123 Saarbrücken, Germany

ingenieurwissenschaften
htw saar

Einleitung

- Pulmonaler Mukus ist ein komplexes, heterogenes Polymer-Gerüst
- Es dient als biologische Barriere zwischen Atemluft und Epithelzellen in den Hauptbronchien
- Mukus kann als Gerüst aus großen Glycoproteinen, den sogenannten Mucinen, beschrieben werden
- Die Zwischenräume zwischen den Mucinen sind mit einem niedrig-viskosen Interstitial-Fluid gefüllt
- Mukus ist ein neues Ziel zur Applikation von Wirkstoff-Trägern



Modell

- Mukus kann als drei-dimensionale, isotrope Zellstruktur mit eingeschränkter Geometrie beschrieben werden
- Die Diffusion der Partikel in dieser Struktur im Interstitial-Fluid wird als stochastischer Bewegungsprozess von Punkt-Partikeln mithilfe von MATLAB numerisch simuliert und analytisch gelöst
- Implementierung der masselosen Smoluchowski-Annäherung der Langevin-Gleichung für überdämpfte Partikel mit dem Diffusionskoeffizienten aus der Stokes-Einstein-Beziehung [2]
- Lösung der Differentialgleichung mithilfe der Euler-Maruyama-Methode und Diskretisierung

$$D_0 = \frac{k_B T}{6\pi\eta R} \quad \frac{d}{dt} x = \sqrt{2D_0 \Delta t} g_r \quad \left. \begin{array}{l} MSD(\tau) = \langle (x(\tau) - x(0))^2 \rangle \\ D_{eff}(\alpha) = \frac{MSD(\tau)}{2n\tau^\alpha} \end{array} \right\}$$

$$x_{k+1} = x_k + \sqrt{2D_0 \Delta t} g_r$$

- Berechnung der mittleren quadratischen Verschiebung (MSD) der Partikel und des effektiven Diffusionskoeffizienten mit dem Anomalie-Exponent α (entspricht der Steigung von $\log MSD$ vs. $\log Zeit$)
- Abstrahierung des vereinfachten geometrischen Modells mit Zellgröße A und "Fenster"-Größe d zu reduzierter Zellgröße $L = A/2R$ und semi-permeablen Membranen mit Übergangswahrscheinlichkeiten und reflektierenden Wänden

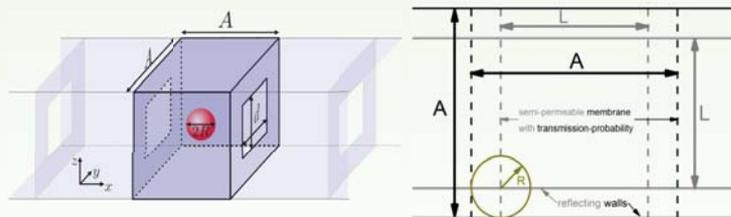


Abb.2: Visualisierung der geometrischen Modelle

Ergebnisse

- Einführung der Reflektionswahrscheinlichkeit r nach Singer [3] und der analytischen Lösung für komplett eingeschränkte Partikeldiffusion nach Kusumi [4] zur Entwicklung eines analytischen Modells

$$r = 1 - P_M \sqrt{\Delta \tau} \longrightarrow (1-r) MSD(\tau) = 2D_0 \tau +$$

$$(r) MSD(\tau) = \frac{L^2}{6} - \frac{16L^2}{\pi^4} \sum_{m=1(\text{odd})}^{\infty} \frac{1}{m^4} \exp\left\{-\frac{1}{2} \left(\frac{m\pi\sqrt{2D_0}}{L}\right)^2 \tau\right\}$$

Ergebnisse

- Eine weitere Entwicklung des analytischen Modells beinhaltet das Verhältnis aus theoretischem (Stokes-Einstein-Beziehung) und effektivem Diffusionskoeffizienten

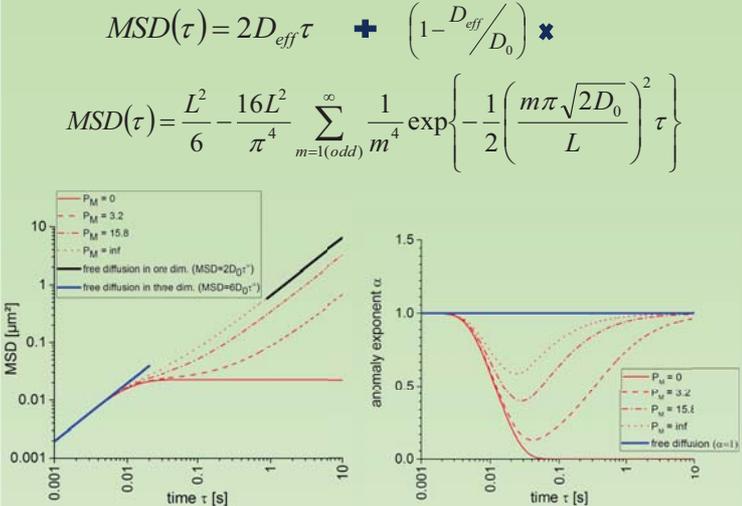


Abb.3: MSD(τ) (links) und Anomalie-Exponent α (rechts) für verschiedene P_M -Werte

$$MSD(\tau) = \begin{cases} 2nD_0\tau^\alpha & \text{mit } \alpha = 1, \tau \rightarrow 0, 0 \leq r \leq 1 \\ 2nD_{eff}\tau^\alpha & \text{mit } \alpha = 1, \tau \rightarrow \infty, 0 < r < 1 \\ 3L^2/2n & \text{mit } \alpha = 0, \tau \rightarrow \infty, r = 1 \\ 2nD_0\tau^\alpha & \text{mit } \alpha = 1, \tau \rightarrow \infty, r = 0 \end{cases}$$

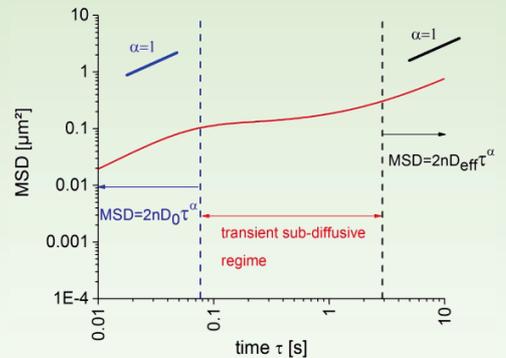


Abb.4: MSD(τ) mit den verschiedenen diffusiven Regimen

- Diffusives Regime für sehr kleine ($\sim D_0$) und sehr große ($\sim D_{eff}$) Zeiten mit $\alpha = 1$
- Transientes sub-diffusives Regime mit $\alpha < 1$

Zusammenfassung

- Trotz der Beziehung $D_0 \sim 1/R$ wurden geringe Diffusivitäten beobachtet, die auf topologische Eigenschaften des Mucus zurückzuführen sind
- Das beobachtete sub-diffusive Regime ist **transient**, daher handelt es sich bei dem Transportmechanismus von Partikeln in Mukus nicht um eine reale Subdiffusion, sondern er ist partiell eingeschränkt aufgrund der heterogenen Mukus-Struktur
- Eine freie Diffusion mit D_0 für sehr kleine und mit D_{eff} für sehr große Zeitskalen wurde beobachtet
- Eine Annäherung an experimentelle Daten und Vorhersage der Diffusion in sehr kleinen und sehr großen Zeitskalen ist unter bestimmten Voraussetzungen möglich

Referenzen

- [1] Kirch, J.; Schneider, A.; Abou, B.; Hopf, A.; Schäfer, U. F.; Schneider, M.; Schall, C.; Wagner, C., and Lehr, C.-M. (2012), 'Optical Tweezers reveal relationship between microstructure and nanoparticle penetration of pulmonary mucus', PNAS 109, 18355-18360.
- [2] Burada, P. S.; Hänggi, P.; Marchesoni, F.; Schmid, G., and Talkner, P. (2009), 'Diffusion in Confined Geometries', ChemPhysChem 10, 45-54.
- [3] Singer, A.; Schuss, Z.; Osipov, A., and Holcman, D. (2008), 'Partially Reflected Diffusion', J. Appl. Math. 68, 844-868.
- [4] Kusumi, A.; Sako, Y., and Yamamoto, M. (1993), 'Confined Lateral Diffusion of Membrane Receptors as Studied by Single Particle Tracking (Nanoavid Microscopy). Effects of Calcium-induced Differentiation in Cultured Epithelial Cells', Biophysical Journal 65, 2021-2040.

Kontakt

matthias.ernst@htwsaar.de



Numerische Simulation von 2D Wasserströmungen mit freien Rändern

M. Kilgenstein & J.-M. Sautter

Labor für Scientific Computing und Simulation, Hochschule Aschaffenburg

Zusammenfassung

Die Anforderungen an die Simulation von Strömungen haben insbesondere im Bereich der Computergraphik sowie auch im Bereich der Optimierung industrieller Anwendungen stark zugenommen. Entsprechend groß ist daher der Bedarf an neuen, effizienten Algorithmen.

Die im Rahmen des Projekts entwickelte Software ermöglicht die Simulation zweidimensionaler, instationärer, inkompressibler, viskoser Strömungen Newton'scher Fluide und erweitert [4] um zeitlich variable Fluidgebiete mit freien Rändern. Die Software erlaubt auch Fluidgebiete, deren Topologie sich zeitlich ändert, wie dies beispielsweise bei einem in eine Pfütze fallenden Regentropfen der Fall ist. Die Implementierung erfolgte in MATLAB.

Einleitung

Numerische Verfahren zur Lösung sogenannter freier Randwertprobleme spielen in vielen Problemen der Natur- und Ingenieurwissenschaften eine große Rolle. Die Anforderungen an die realitätsnahe Darstellung von z. B. Wasserströmungen haben insbesondere im Bereich der Computergraphik zugenommen. Die Bedeutung dieser Verfahren, die Anzahl und Komplexität der simulierten Szenarien sind in den letzten Jahren stark gewachsen.

Mit der Genauigkeit der Visualisierung steigen auch der Speicherbedarf und die Berechnungszeit immens an. Deswegen gilt es, immer effizientere Algorithmen zur Simulation von Strömungen mit freien Oberflächen zu entwickeln.

Eine Vielzahl von realen Strömungsphänomenen können durch freie Randwertprobleme beschrieben werden. Darunter zählen in der Natur beispielsweise die Dynamik von Meereswellen, ein Regentropfen, der in eine Pfütze fällt oder die Überschwemmung eines Flusstals aufgrund eines Dammbrochs.

Wichtige Anwendungen von freien Randwertproblemen in der Industrie sind Spritzgussverfahren zur Herstellung von Plastikbauteilen (vgl. [2, 6]) oder Beschichtungsprozesse (vgl. [1]). Auch Mehrphasenströmungen, bei denen die Grenze zwischen mindestens zwei verschiedenen Fluiden den freien Rand darstellt, gehören zu dieser Kategorie von Strömungen.

Mathematisches Modell

Ebene, instationäre, inkompressible, viskose Strömungen Newton'scher Fluide in einem Gebiet Ω_t , wie beispielsweise Wasserströmungen, werden durch die Navier-Stokes Gleichungen

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} &= \nu \Delta \mathbf{u} - \nabla p & \text{in } \Omega_t \\ \nabla \cdot \mathbf{u} &= 0 & \text{in } \Omega_t \end{aligned}$$

beschrieben. Dabei bezeichnen $\mathbf{u} : \mathbb{R} \times \mathbb{R}^2 \rightarrow \mathbb{R}^2$ mit $(t, \mathbf{x}) \mapsto \mathbf{u}(t, \mathbf{x})$ die Geschwindigkeit, $p : \mathbb{R} \times \mathbb{R}^2 \rightarrow \mathbb{R}$ den Druck und $\nu \in \mathbb{R}$ die kinematische Viskosität des Fluids.

Bei instationären Strömungen mit freien Oberflächen ist das Gebiet Ω_t zeitabhängig. Zur Lösung der Navier-Stokes Gleichungen werden auf $\partial\Omega_t$ Randbedingungen an festen und freien Rändern sowie an Rändern mit Ein- und Ausfluss des Fluidgebiets benötigt. Die Randbedingungen an freien Rändern ergeben sich aus der Oberflächenspannung

$$\sigma \mathbf{n} = \gamma \frac{1}{r} \mathbf{n}$$

Dabei ist σ der Spannungstensor, γ der materialabhängige Koeffizient für die Oberflächenspannung, \mathbf{n} der Normalenvektor und r der Krümmungsradius des freien Rands. In den folgenden Simulationen ist $\gamma = 0$, d. h. sie basieren auf der Annahme vernachlässigbarer Oberflächenspannung.

Beschreibung der freien Oberfläche

Die Verfolgung der freien Oberfläche erfolgt mit Hilfe der Marker-and-Cell (MAC) Methode [3, 5], bei der masselose Hilfspartikel aufgrund der aktuellen Fluidgeschwindigkeit mit der Strömung bewegt werden. Aufgrund der Positionen der Hilfspartikel wird das Strömungsgebiet zunächst in innere Fluidzellen, Oberflächenzellen und leere Zellen unterteilt (siehe Abb. 1). Das Fluidgebiet wird durch die Menge der Fluidzellen und der freie Rand durch die Menge der Oberflächenzellen approximiert.

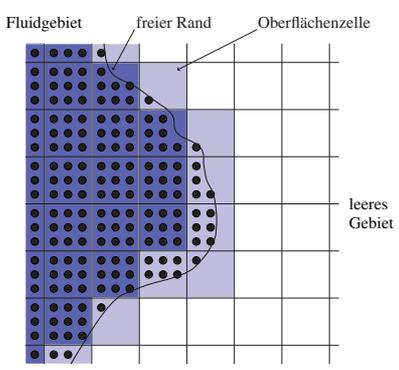


Abb. 1: Einteilung des Gebiets mit Hilfe von Hilfspartikeln

Eine Oberflächenzelle ist eine Fluidzelle, die dadurch charakterisiert wird, dass sie in Richtung der Koordinatenachsen an mindestens eine leere Zelle angrenzt. In jedem Zeitschritt erfolgt die Einteilung der Zellen durch die Positionen der Hilfspartikel. Befinden sich in einer Zelle Hilfspartikel, gehört diese Zelle zum Fluidgebiet und andernfalls zum leeren Gebiet.

In einem weiteren Schritt werden die Oberflächenzellen in verschiedene Typen weiter aufgeteilt, für die jeweils unterschiedliche Randbedingungen am freien Rand gesetzt werden müssen (siehe Abb. 2). Innere Fluidzellen sind in Abb. 2 hellrot und mit einem „F“ und leere Zellen dunkelrot und mit einem „L“ markiert. Die übrigen Zellen sind verschiedene Oberflächenzellen, die die Buchstaben der Himmelsrichtungen tragen, in denen sie an leere Zellen angrenzen, beispielsweise „N“ für Nord.

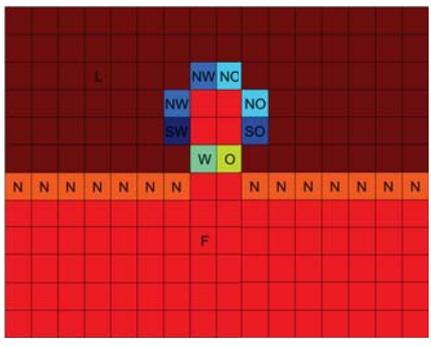


Abb. 2: Oberflächenzellen beim fallenden Tropfen

In Abb. 3 werden die Hilfspartikel bei der Simulation eines brechenden Staudamms (vgl. [1, 3]) dargestellt. An manchen Stellen des Gebiets, an denen sich (physikalisch) Fluid befindet, kann es vorkommen, dass sich dort keine Hilfspartikel befinden. Dies kann zur Folge haben, dass unphysikalische leere Zellen bzw. Oberflächenzellen innerhalb des eigentlichen Fluidgebiets entstehen.

Daher werden die Hilfspartikel im inneren Fluidgebiet in regelmäßigen Zeitabständen neu angeordnet, sodass die Hilfspartikel im inneren Fluidgebiet gleichmäßig verteilt sind. Durch diese Maßnahme kann auch die Anzahl der Hilfspartikel pro Zelle reduziert und somit die Berechnungszeit deutlich verringert werden.

In Abb. 4 sind die Ergebnisse der Simulation einer Strömung über eine Stufe mit freier Oberfläche dargestellt.

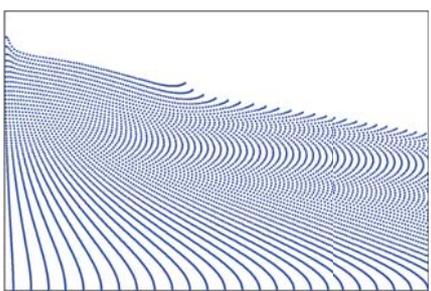


Abb. 3: Hilfspartikel beim brechenden Staudamm

Simulationsergebnisse

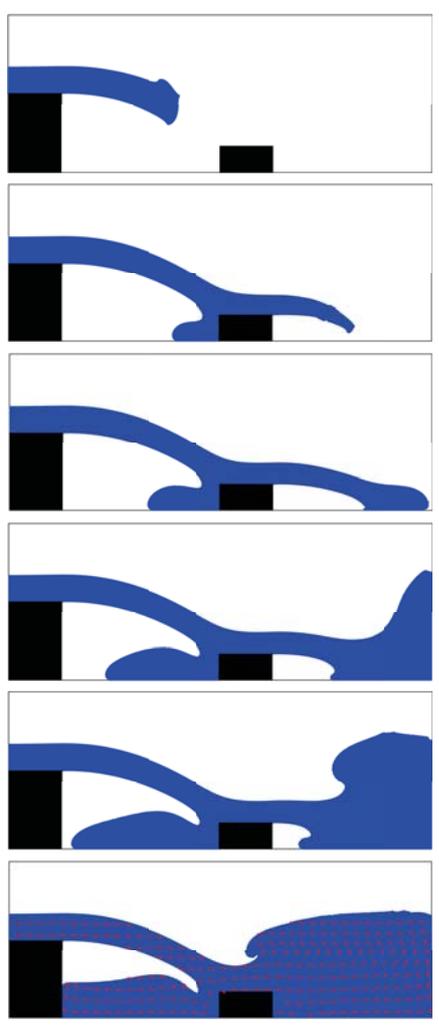


Abb. 4: Strömung über eine Stufe mit freier Oberfläche

Fazit

Neben den hier dargestellten Simulationsergebnissen konnten die im Rahmen des Projekts entwickelten und in MATLAB implementierten Algorithmen auch an weiteren verschiedenartigen Strömungsphänomenen eindrucksvoll demonstriert werden.

Literatur

- [1] M. Griebel, T. Dornseifer und T. Neunhoeffer. *Numerical Simulation in Fluid Dynamics – A Practical Introduction*. SIAM, 1997.
- [2] A. Grillet u. a. „Numerical analysis of flow mark surface defects in injection molding flow“. In: *Journal of Rheology* 46.3 (2002).
- [3] F. H. Harlow und J. E. Welch. „Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface“. In: *The Physics of Fluids* 8.12 (1965).
- [4] M. Kilgenstein, K. M. Kraenzel, S. Schuster, M. Stühmer und J.-M. Sautter. *Ein kompakter und vektorisierter MATLAB Code zur Lösung der inkompressiblen Navier-Stokes Gleichungen*. MATLAB Expo. München, 2015.
- [5] S. McKee, M. F. Tomé, V. G. Ferreira, J. A. Cuminato, A. Castelo, F. S. Sousa und N. Mangiacavchi. „The MAC method“. In: *Computer & Fluids* 37 (2008), S. 907–930.
- [6] E. Mitsoulis. „Computational Polymer Processing“. In: *Modeling and Simulation in Polymers*. Hrsg. von P. D. Gujrali und A. I. Leonov. Wiley-VCH, 2010. Kap. 4, S. 127–195.

Contactless Detection of Fluid Turbulences with a Millimeter Wave FMCW-Radar

B. Hattenhorst, C. Baer, and T. Musch

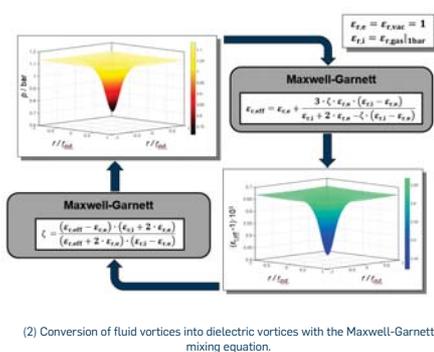
Motivation

The industrial volume and flow measurement has a great importance in process plants. With its help, process chains can be optimized in many different aspects. However, flow measurements face a huge challenge in extreme environments like flare gas plants, where the gaseous medium is conveyed with very high velocities and the flow should not be influenced by the measurement. In this scenario, state-of-the-art flow measurements fail. This is where the contactless, millimeter-wave-based flow measurement approaches.



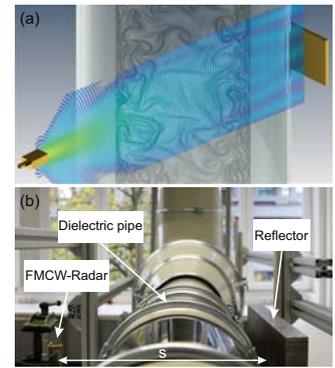
(1) Gas flare: <http://www.trinityconsultants.com>

Measurement Principle



- Fluid vortices have a pressure distribution referring to Rankine, Oseen, and others
- For a single ingredient the total pressure of a gas mixture under atmospheric conditions is equal to the volume fraction
- The Maxwell-Garnett equation for dielectric mixing provides an effective relative permittivity $\epsilon_{r,eff}$ for fluid vortices
- Disparity between reference $\epsilon_{r,ref}$ and vortex measurement leads to transmission phase fluctuations (TPF):

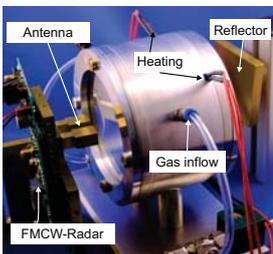
$$\Delta\varphi = 360^\circ \cdot \left(\sqrt{\epsilon_{r,eff}} - \sqrt{\epsilon_{r,ref}} \right) \cdot \frac{f_c}{c_0} \cdot 2s$$



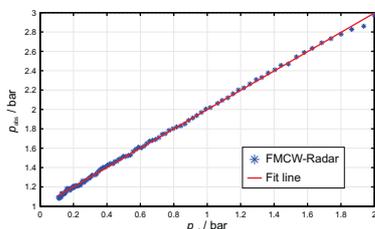
(3) (a) Simulative measurement principle, (b) measurement setup in laboratory environment.

Contactless Pressure Measurement

- Pseudo transmission phase measurements with pressure chamber
- Correlation between pressure sensor values and evaluated phase
- Phase results can be translated into the absolute pressure



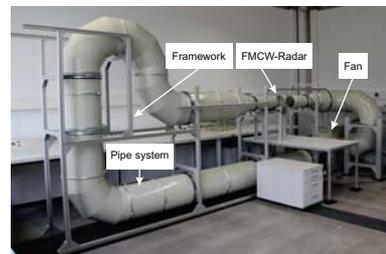
(4) Contactless pressure measurement setup with gas chamber, FMCW-Radar, and reflector.



(5) Phase measurement results translated into absolute pressure values over the relative pressure values of a conventional pressure sensor.

Measurement Setup

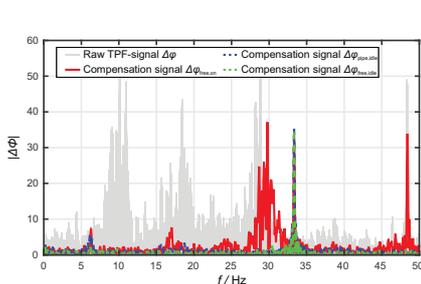
- **Challenge:**
Vibrations of pipe systems



(6) Measurement setup consisting of a pipe system that is driven by a fan and held by a framework.

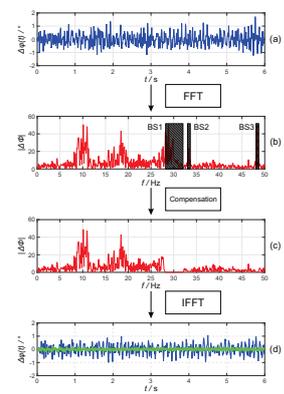
- **Dimensions:**
 - Height: 2.30 m
 - Length: 5.20 m
 - Width: 1.50 m
- **Pipes:**
 - max. 400 mm Ø
 - min. 200 mm Ø
- **Flow:**
 - max. 11100 m³/h
 - max. 80 m/s

Compensation Measurements



(7) Spectra of the measured TPF-signal superimposed with three radar compensation measurements.

- Compensation measurements can be done to cancel out disturbances:
 1. Radar and reflector are placed beside the pipe with fan in on-state
 2. Radar and reflector are placed beside the pipe with fan on idle-state
 3. Pipe between radar and reflector with fan in idle-state
- The compensation measurements include disturbances of the radar system and the vibrations of the setup
- To cancel out the spurious signal components in a raw data measurement, band-stop and notch filters are used
- Filtered signal depends on processes inside the pipe



Evaluation of HS-GC-IMS fingerprints of *Pyricularia oryzae* with MATLAB

A. J. K. Roth, M. Lerch, P. Weller*

Institute for Instrumental Analytics and Bioanalytics, Mannheim University of Applied Sciences
Paul-Wittsack-Str. 10, 68163 Mannheim, Germany, p.weller@hs-mannheim.de

1. Introduction

Determination of the mode of action (MoA) is an important part of the development of new agents in the agrochemical field and also the basis for the comprehension of resistances against crop protection products. Up to now, more than 40 MoA for fungicides are known.^[1] The determination of these MoA often employs complex and time consuming metabolomic studies.^[2] We are currently developing a fast new method for the determination of MoA of fungicides by headspace gas chromatography coupled with ion mobility spectrometry (HS-GC-IMS) on the example of the rice blast pathogen *Pyricularia oryzae* (Fig. 2 and 3). The method is based on the comparison of microbial volatile organic compound (MVOC) profiles of treated and untreated mycelium supported by multivariate tools such as PCA or Parafac.^[3] Data preprocessing as well as multivariate analysis will be performed with MATLAB software.



Fig. 1: *P. oryzae* on rice, in agar and liquid culture

2. Working Principle and Sample Preparation

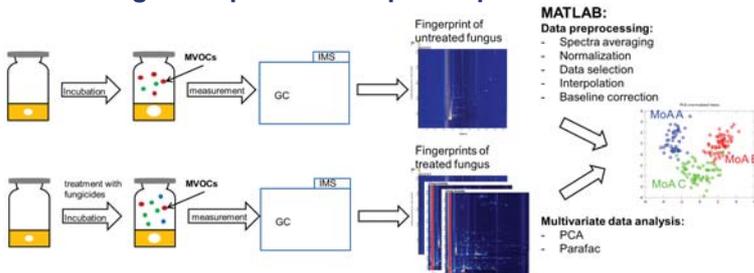


Fig. 2: Principle of fungicide mode-of-action-fingerprinting by HS-GC-IMS

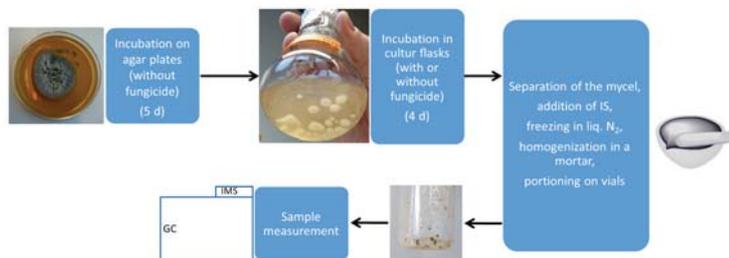


Fig. 3: Sample preparation workflow

3. Results

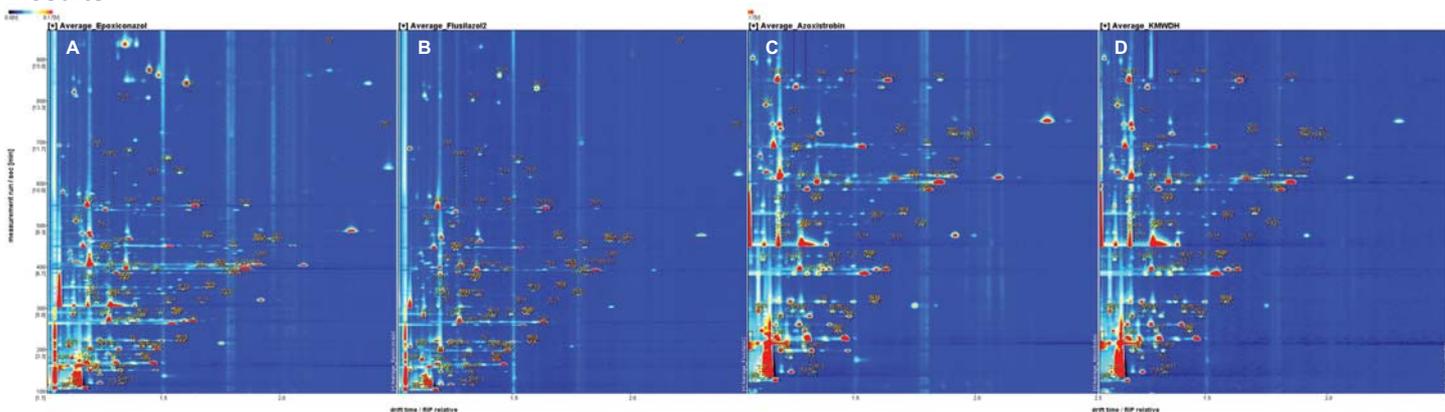


Fig. 4: MVOC-fingerprints of *P. oryzae*: treated with epoxiconazole (A), treated with flusilazole (B), treated with azoxistrobin (C, MoA-group G1), treated with kresoxim-methyl (D, MoA-group G1).

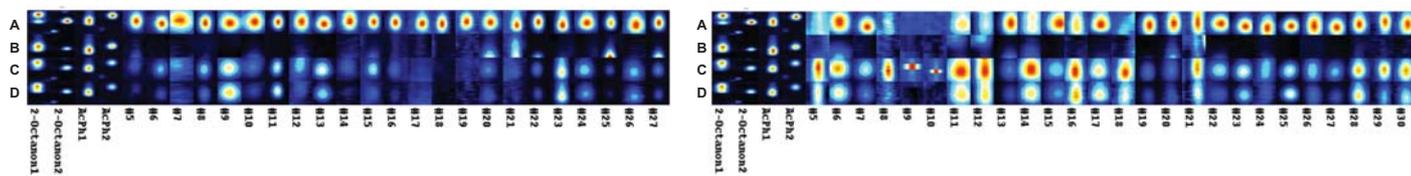


Fig. 5: Gallery view of selected peaks: *P. oryzae* treated with epoxiconazole (A, MoA-group C3), treated with flusilazole (B, MoA-group C3), treated with azoxistrobin (C, MoA-group G1), treated with kresoxim-methyl (D, MoA-group G1).

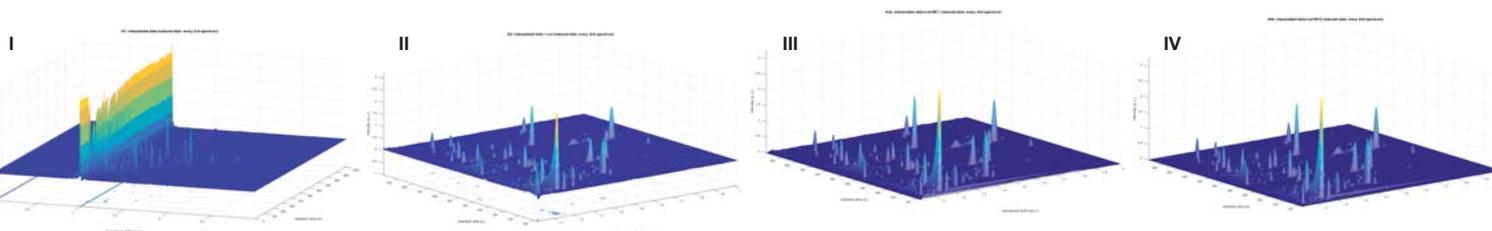


Fig. 6: Visualization of data preprocessing steps with MATLAB: Fingerprint data after averaging and interpolation (I), data selection (II), baseline correction (rubberband) in retention time dimension (III), baseline correction (rubberband) in drift time dimension (IV).

4. References

- [1] Fungicide Resistance Action Committee, <http://www.frac.info/moa-expert-panel>
- [2] K. Grossmann, N. Christiansen, R. Looser, S. Tresch et al., *Pest Manag. Sci.* **2012**, *68*, 494–504.
- [3] R. Bro, *Chemometrics and Intelligent Laboratory Systems*, **1997**, *38*, 149–171.

5. Acknowledgement

We thank the Federal Ministry for Economic Affairs and Energy for financial support and the Gesellschaft für analytische Sensorsysteme m. b. H., Dortmund, for material support (IMS-module).



Application of MATLAB in quality assessment of olive oil using orthogonal non-targeted fingerprint analysis and chemometrics

N. Gerhardt ^{1,3}, P. Weller ^{1,3}, S. Rohn ²

¹ Institute for Instrumental Analytics and Bioanalysis, Mannheim University of Applied Sciences, Germany, p.weller@hs-mannheim.de

² Institute of Food Chemistry, Hamburg School of Food Science, University of Hamburg, Germany

³ Center for Applied Research in Biomedical Mass Spectrometry (ABIMAS), Hochschule Mannheim

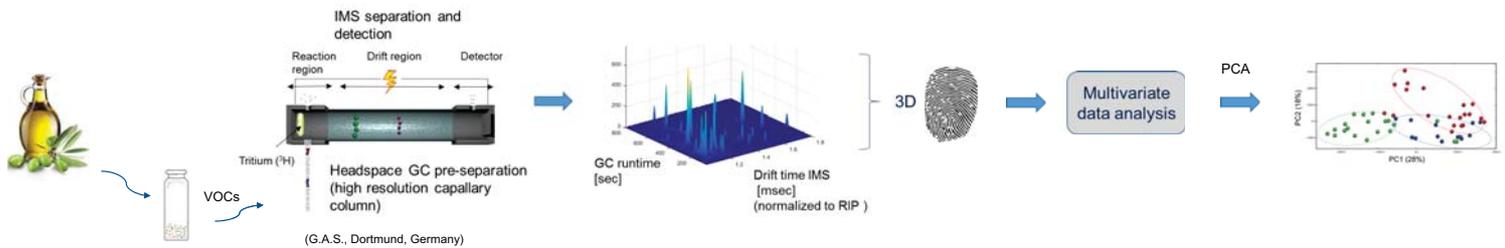
Introduction

The evaluation of extra virgin olive oil (EVOO) with respect to quality and authenticity has been in the focus of a plethora of studies in the past years and still poses a challenge for researchers. EVOO is considered the highest quality of the three different quality grades and must meet certain guidelines in order to hold this title. Other lower-quality olive oils are known as virgin olive oil (VOO) and lampante olive oil (LOO). In the most common type of commercial fraud of olive oil, either the geographical indications are used in a misleading manner or a bottle labeled "extra virgin" may not always be an authentic EVOO. Thus, in particular, sensitive, rapid and low-cost analytical methods are highly relevant for enforcement purposes to ensure EVOO authenticity.

Non-targeted headspace analysis of volatile organic compounds (VOCs) by high resolution gas chromatography (HRGC), coupled to ion mobility spectrometry (IMS) provides a fast and effective approach for the discrimination of EVOOs of different geographical origins or for olive oil of different quality categories. This is especially due to minimal sample preparation and short run times, paired with the orthogonal separation power of retention time vs. drift time in GC-IMS.

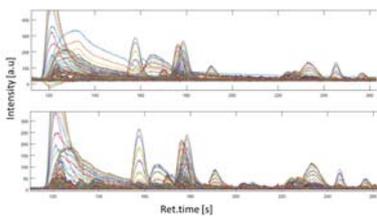
The resulting product-characteristic three-dimensional fingerprints are evaluated by multivariate statistics tools implemented in custom MATLAB routines.

Workflow Concept

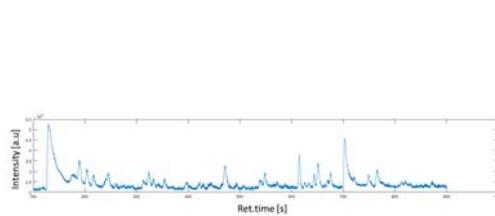


Preprocessing steps

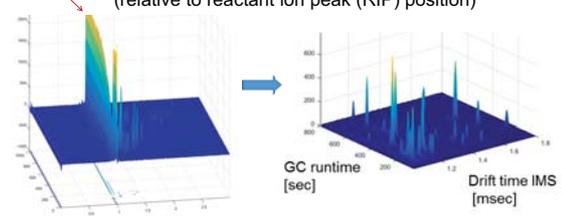
A: Baseline correction (rubberband)



B: Cubic spline data interpolation (Total Ion Chromatogram)

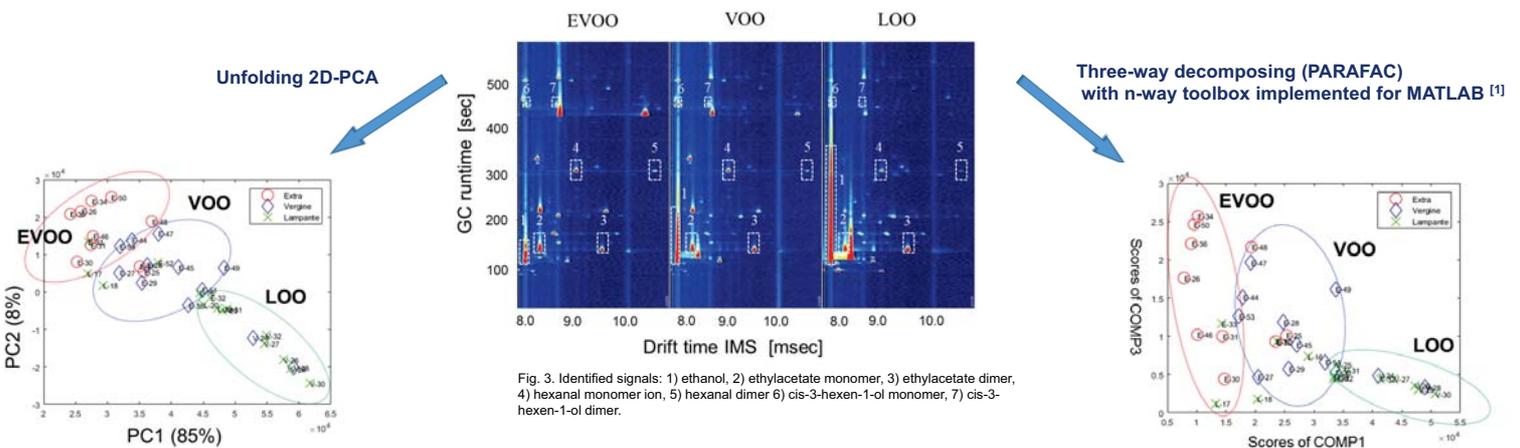


C: Spectral region selection and Normalization (relative to reactant ion peak (RIP) position)



Results

Typical HRGC-IMS chromatogram-mobility plot of the three olive oil quality categories: Extra Virgin, Virgin and Lampante olive oil



Headspace high resolution capillary gas chromatography (HS-HRGC) coupled to IMS in combination with intelligent mathematical algorithms provide a good approach for a non-targeted separation of olive oil samples of different qualities

References

[1] C. A. Andersson, R. Bro, Chemom. Intell. Lab. Syst. 52 (2000) 1-4

Acknowledgements

The authors would like to thank the company Gesellschaft für analytische Sensortysteme (G.A.S., Dortmund, Germany) for providing instrumental support and to the Department of Analytical Chemistry of the University Cordoba (Spain) for supplying olive oil samples.



Strategies to classify crispy sounds and mechanics using dynamic spectral information

Solange Sanahuja, Heiko Briesen

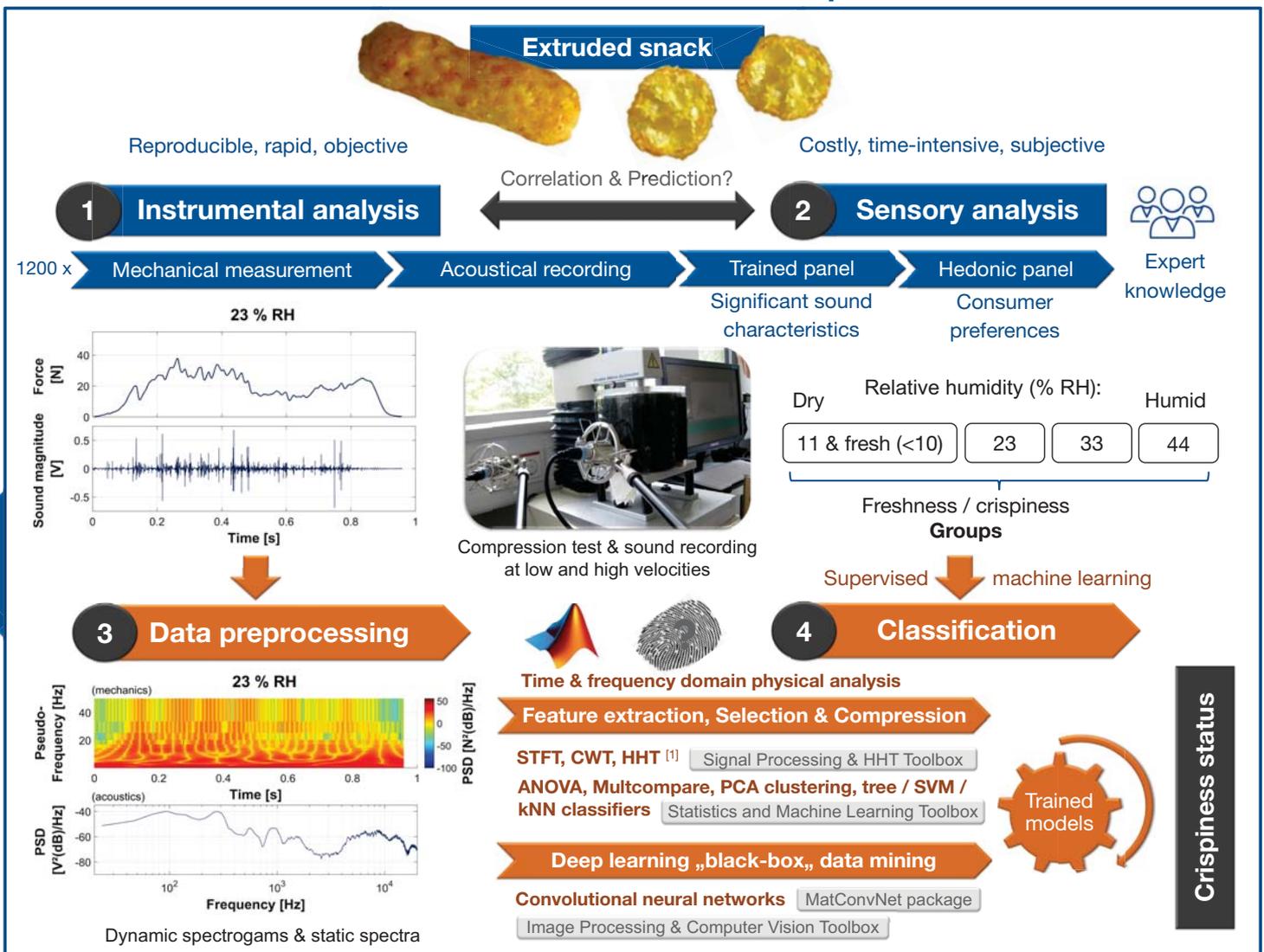
Motivation

- Crispy food textures = stimulating, fresh, pleasant & have → highest impact on consumer preference and quality evaluation [1]
- Food industry and research requirements for quality control and development
- **Crispiness evaluation = essential but causes persistent difficulties in practice, in particular for differentiating low-humidity crispness levels (10 to 20 % RH) [1]**

Goals

- Optimizing crispy products = appropriately stiff and brittle during chewing & release pleasant rhythmic sounds of particular pitch and loudness avoiding use of sensory panels
- Improvement of available texture measurements and data analysis methods
- **Freshness levels classification from instrumental data corresponding to sensory crispiness grades, mimicking multisensory & temporal integration during oral breakdown**

Multimodal texture characterization process



Conclusion

- **Classification needs more than simple mechanical features**
- Instrumental crushing sounds → perception of food freshness
- Dynamic spectral analysis → mathematical analytic description & display of whole complexity of foods' signature
- Multimodal classification → multitude of modern methods & optimum not straightforward, but improved accuracy

Further Improvements

- Synchronized denoising & bone-conducted sounds using transfer function models of isolation box and human head

[1] Sanahuja, S. and Briesen, H. 2015. Dynamic spectral analysis of jagged mechanical signatures of a brittle puffed snack. Journal of Texture Studies, 46, 171-186.

Data Analysis in Personalized Medicine

Segmentation, Classification and Clustering of DICOM-Files using MATLAB

M. Stich¹, K. Schuller¹, S. Seemann¹, J. Vogt¹, M. Lindner¹, R. Ringler¹

¹) Medical Engineering, X-Ray & Molecular Imaging Lab., IfMZ, Ostbayerische Technische Hochschule Amberg-Weiden, Weiden, Germany

Introduction and Motivation

The personalized medicine gains in its importance due to present worldwide demographic changes. The concept of the personalized medicine is to provide diagnoses and therapies adopted to a single human individual. This demographic trend is seen particularly in the field of medical imaging methods of diagnostic radiology and nuclear medicine. With advanced and hybride imaging procedures, for example with the gamma camera, the SPECT and PET of the nuclear medicine treatment and CT or MRI of the traditional x-ray diagnosis, a better understanding of disease, treatment progression, the underlying anatomy, and the physiological and biochemical structures and processes in the human body can be visualized. The computerized image segmentation and classification of tissue structures is a fundamental task which is set down development and improvement in the analysis of image data. In this work, some of the most powerful segmentation and classification methods have been applied to image data sets of different modalities and relevant issues in the field of personalized medicine. The result in terms of quality was therefore quantitatively analyzed and evaluated.

Material and Methods

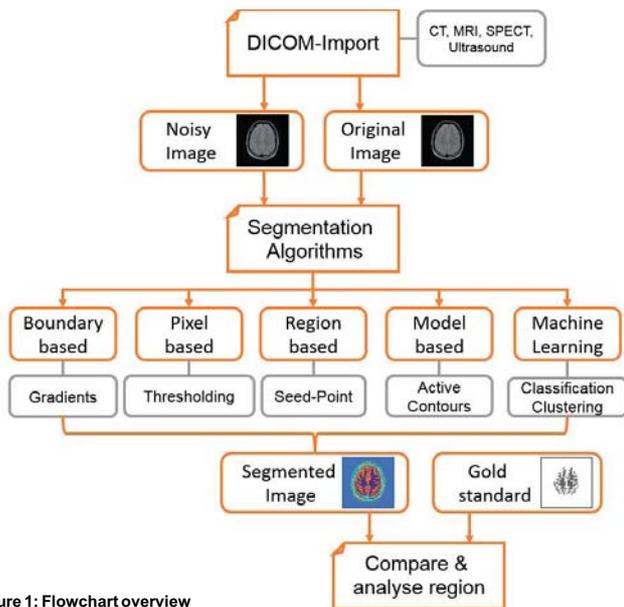


Figure 1: Flowchart overview

A broad range of optimized segmentation, classification and clustering algorithms for medical image segmentation were implemented using MATLAB R2015a. A variety of toolboxes, e.g. image processing toolbox, computer vision toolbox, statistics and machine learning toolbox were used. Moreover, the algorithms were evaluated after applying gaussian noise with five different intensity levels to the image. The segmentation result was compared to a gold standard, manually segmented image. The results are shown in figure 2 and table 1.

Conclusion and Discussion

- Machine learning algorithms (e.g. kNN, kmeans) show good segmentation results for noisy images in comparison to other methods.
- The results for segmentation depend on the anatomical structure.
- Segmentation with Watershed algorithm is not suitable for high noise levels.
- There is nearly no correlation between SNR or CNR and the segmentation.

Results

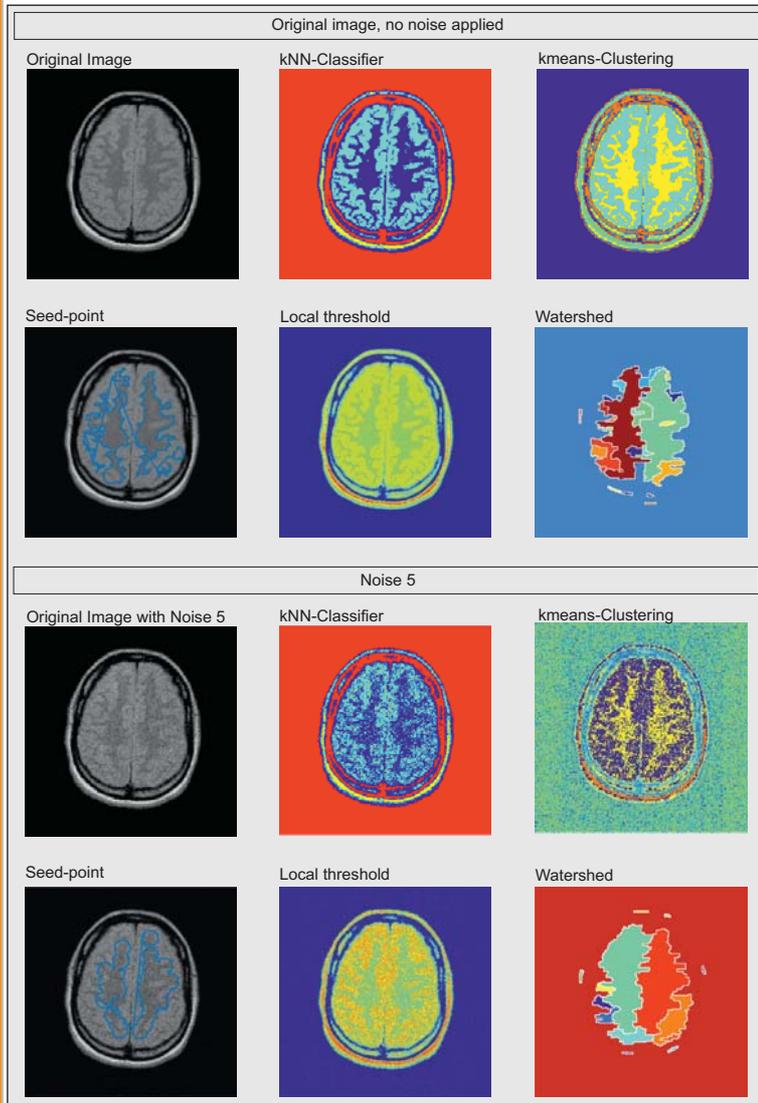


Figure 2: A variety of segmentation algorithms applied to a MRI brain image

Table 1 shows the quantitative results for the MRI brain image with the decreasing signal-to-noise (SNR) and contrast-to-noise ratio (CNR).

	MRI-brain DICOM					
	Original image	Noise 1 mean = 0.0 std = 0.0005	Noise 2 mean = 0.0 std = 0.001	Noise 3 mean = 0.0 std = 0.0015	Noise 4 mean = 0.0 std = 0.002	Noise 5 mean = 0.0 std = 0.0025
SNR	20,35	19,74	18,90	18,32	17,73	17,18
CNR	20,59	17,64	15,61	14,41	13,09	12,25
Golden standard						
Grey matter [pixel]	8272					
Deviation [pixel]	0					
Deviation [%]	0,00					
kNN - Segmentation						
Grey matter [pixel]	8246	9150	8467	8615	8607	9131
Deviation [pixel]	-26	878	195	343	335	859
Deviation [%]	-0,31	10,61	2,36	4,15	4,05	10,38
kMeans - Segmentation						
Grey matter [pixel]	8194	8921	8859	8355	8587	8622
Deviation [pixel]	-78	649	587	83	315	350
Deviation [%]	-0,94	7,85	7,10	1,00	3,81	4,23
Seed-point - Segmentation						
Grey matter [pixel]	8034	9923	9223	9474	9517	9235
Deviation [pixel]	-238	1651	951	1202	1245	963
Deviation [%]	-2,88	19,96	11,50	14,53	15,05	11,64
Local threshold (advanced) - Segmentation						
Grey matter [pixel]	8112	8446	8521	7384	8177	7029
Deviation [pixel]	-160	174	249	-888	-95	-1243
Deviation [%]	-1,93	2,10	3,01	-10,74	-1,15	-15,03
Watershed - Segmentation						
Grey matter [pixel]	7347	7723	7873	6311	7668	8867
Deviation [pixel]	-925	-549	-399	-1961	-604	595
Deviation [%]	-11,18	-6,64	-4,82	-23,71	-7,30	7,19

Table 1: SNR, CNR and quantitative segmentation results of the MRI brain image.

Einleitung

- Optimierung von Hochleistungs-pin-Dioden umfasst zunehmend anspruchsvollere Produktionsschritte wie lokale Trägerlebensdauereinstellung oder tiefe Feldstopps [1]
- Untersuchung des dynamischen Verhaltens von Bauelementen als Teil eines Schaltkreises \Rightarrow Entwicklung von Modellen notwendig, die in Schaltungssimulatoren integrierbar sind
- Implementierung in MATLAB: kompakte Schreibweise durch Matrix-Vektor Notation, Möglichkeit der weiteren Untersuchung des Modells mit PLECS in Simulink

Modell mit dynamischem Avalanche

- Zentrale Gleichung zur Modellierung des Ladungsträgerplasmas: Ambipolare Diffusionsgleichung (ADG):

$$D_A \frac{\partial^2 p(x,t)}{\partial x^2} = \frac{\partial p(x,t)}{\partial t} + \frac{p(x,t)}{\tau}$$

- p - Löcherkonzentration, τ - Hochinjektions-Trägerlebensdauer, D_A - ambipolarer Diffusionskoeffizient
- Separation von Speicherladungsgebiet und Raumladungszonen zu jeden Zeitpunkt notwendig \rightarrow Free Boundary Problem mit den Rändern $x_1(t)$ und $x_2(t)$, siehe Abb. 1
- Neumann-Randbedingungen zur Lösung der ADG:

$$\frac{\partial p(x,t)}{\partial x} \Big|_{x_i} = \frac{i_n(x_i(t),t)}{2qAD_n} - \frac{i_p(x_i(t),t)}{2qAD_p} \text{ für } i = 1,2$$

- Dirichlet-Randbedingungen zur Bestimmung des freien Randes:

$$p(x,t)|_{x_1} = N_D^+ \text{ für } t > t_1 \quad \text{and} \quad p(x,t)|_{x_i} = N_D^+, \quad i = 1,2 \text{ für } t > t_2$$

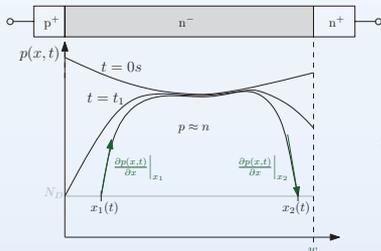


ABB. 1: Notation innerhalb des Speicherladungsgebiets

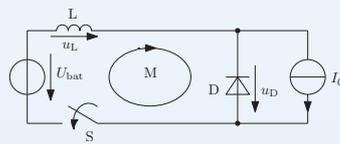


ABB. 2: Kommutierungs-Schaltkreis

- Finite-Elemente-Methode zur Lösung des Problems in einem gegebenen Gebiet

$$D_A \mathbf{K} \cdot \mathbf{p}(t) + \frac{1}{\tau} \mathbf{M} \cdot \mathbf{p}(t) + \frac{d(\mathbf{M} \cdot \mathbf{p}(t))}{dt} = D_A \mathbf{f}$$

- \mathbf{M} - Massenmatrix, \mathbf{K} - Steifigkeitsmatrix, \mathbf{p} - Lösungsvektor, \mathbf{f} - rechte Seite mit Neumann-Randbedingungen
- Linien-Methode [2] zur Bestimmung des freien Randes: Randpunkt springt entlang der Diskretisierung des Startgebietes $[0, w]$; für jeden Schritt wird passender Zeitschritt mit Bisektionsmethode so bestimmt, dass Dirichlet-Randbedingungen erfüllt sind.
- Nach Umstellen der Maschengleichung der Masche M in Abb. 2 und nach Definition der Spannung $u_D(t)$ erhält man

$$i_D(t) = I_F - \frac{U_{bat}}{L} t + \frac{1}{L} \int_0^t u_D(\tilde{t}) d\tilde{t} \quad \text{und} \quad u_D(t) = \int_0^w E(x) dx$$

Kontakt/Literaturangaben

Vanessa Basler +49 89 289 23111 basler@tep.ei.tum.de

- [1] D. Silber et al, "Improved dynamic properties of GTO-thyristors and diodes by proton implantation", 1985 International Electron Devices Meeting, pp. 162-165, 1985.
[2] J. Crank, "Free and Moving Boundary Problems", Oxford University Press Inc., New York, 1984.
[3] R. Siemieniec et al, "Determination of parameters of radiation induced traps in silicon", Solid-State Electronics. vol. 46, pp. 891 - 901, June 2002.

Lokale Trägerlebensdauereinstellung

- Shockley-Read-Hall (SRH)-Lebensdauer für ein Rekombinationszentrum (trap) m ist gegeben durch die SRH-Gleichung

$$\tau_{SRH,m} = \frac{\tau_{p,m}(n_0 + n_{1,m} + \Delta n) + \tau_{n,m}(p_0 + p_{1,m} + \Delta p)}{p_0 + n_0 + \Delta n}$$

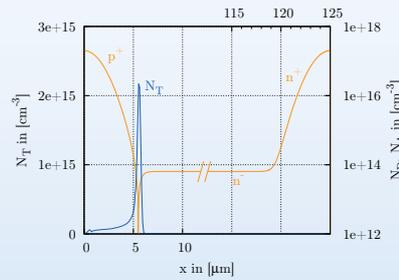


ABB. 3: orange: Dotierprofil, blau: Profil der zusätzlichen Rekombinationszentren

- Falls Rekombinationszentren nicht miteinander wechselwirken: Kombination zu einer effektiven Trägerlebensdauer mit Hilfe der Mathiessen Regel [3] möglich

$$\frac{1}{\tau} = \frac{1}{\tau_{dop}} + \sum_{m=1}^N \frac{1}{\tau_{SRH,m}}$$

Ergebnisse

- Schaltbedingungen: $L = 264 \text{ nH}$, $I_0 = 100 \text{ A}$
- Im Folgenden: Vergleich des Kompaktmodells mit dem Drift-Diffusions-Modell (Simulator Sentaurus Device von Synopsys)

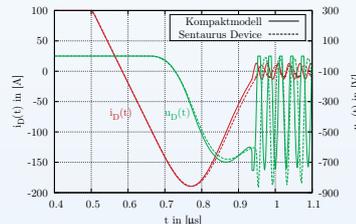


ABB. 4: Strom-Spannungs-Verlauf, $U_{bat} = 325 \text{ V}$

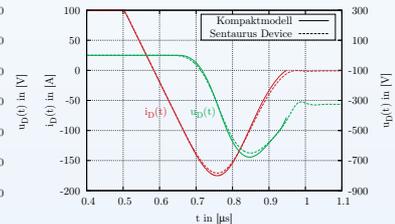


ABB. 5: Strom-Spannungs-Verlauf für Diode mit lokaler Trägerlebensdauereinstellung, $U_{bat} = 325 \text{ V}$

- Durch Rekombinationszentrenpeak am pn-Übergang wird Durchlassladungsträgerverteilung stark abgesenkt
- \Rightarrow Dioden-Schaltverhalten in Abb. 5 ist nun "soft".

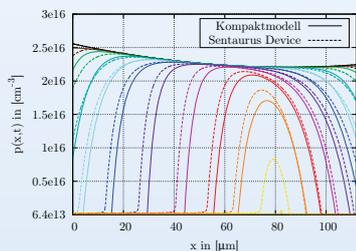


ABB. 6: Ladungsträgerverteilung während des Ausschaltvorgangs, $U_{bat} = 700 \text{ V}$

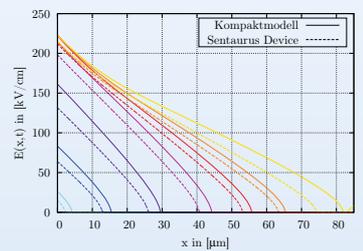


ABB. 7: Elektrische Feldverteilung während des Ausschaltvorgangs, $U_{bat} = 700 \text{ V}$

- Dynamischer Avalanche tritt auf bei Ausschaltvorgängen mit sehr hoher Sperrspannung.
- Zeitliche Entwicklung des Ladungsträgerberges und der Feldverbiegung durch dynamischen Avalanche wird durch das Modell gut wiedergegeben, wie in den Abb. 6 und 7 gezeigt wird.

Zusammenfassung

- Kompaktmodell zeigt gute Übereinstimmung mit dem Diff-Diffusions-Modell
- Lokale Lebensdauereinstellung und dynamischer Avalanche integriert
- Flexibler und hochgradig modularer Modellaufbau
- Modell als physikalisches Bauelement in Schaltungs-Umgebung integrierbar (z.B. PLECS in Simulink)

Abstract

- address the challenge of **natural spontaneous interaction with an automated home environment**
- **distant-speech controlled interaction** with appliances and security services
- **prototype** for scientists and students

1. Example from "Real-Life"



2. Problem Definition

- **natural spontaneous distant-speech interaction**
 - in **noisy and reverberant environments**
 - in case of **multiple speakers and interfering sources**
- use of **MEMS digital microphone arrays**
- employ state-of-the-art algorithms
- develop more accurate and efficient algorithms

3. Contributions and Innovations

- **real-time capable two-room prototype** for simple tasks
- advances in spatio-temporal filtering and keyword spotting
 - beamforming
 - room localization
 - source localization and characterization
 - always-listening mode
 - keyword spotting and wake-up-word recognition

4. Applications

End-User Applications:

- ambient/active assisted living (with elderly people)
- environmental acoustic monitoring, security & surveillance
- interaction with robots
- immersive telepresence

Scientific Applications: versatile research tool for

- acoustic multi-channel signal processing
- distant-speech recognition
- speech understanding
- speaker identification and verification
- spoken dialogue management

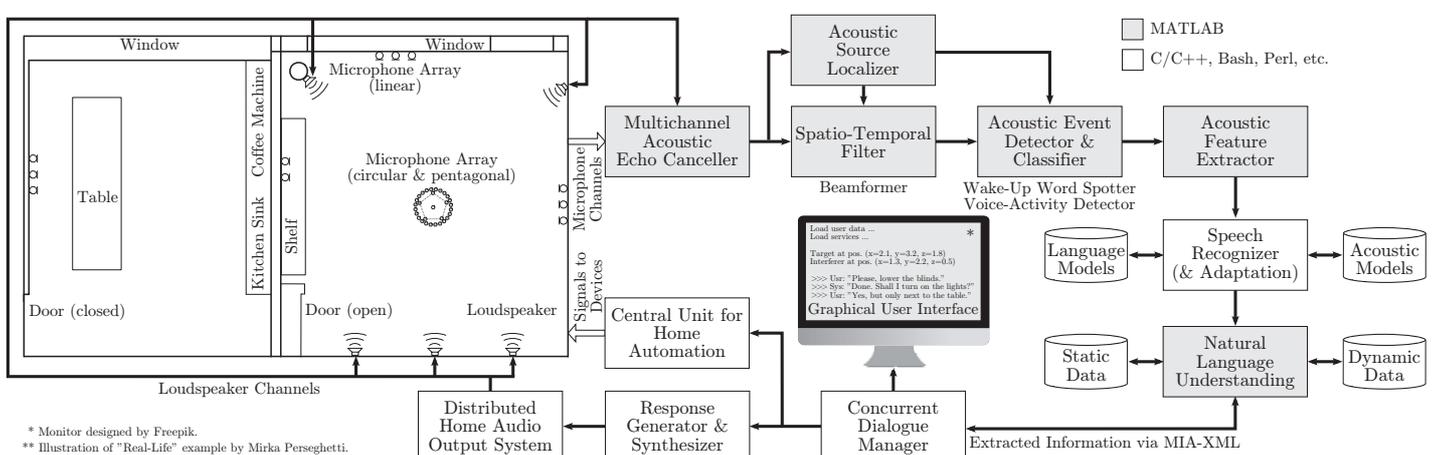
5. "Playrec" Facilitates Real-Time Processing

- free, open-source library to record and play audio signals
- non-blocking access to soundcards using PortAudio
- runs on multiple platforms
- major reason for our system's real-time capability

6. Conclusion

- **two-room prototype** as first proof-of-concept
- **real-time capable** demo at TUGraz
- core modules implemented in MATLAB
- modules easy to exchange, extend, and implement
- test branch for scientists and students
- demo branch for visitors

7. Our System's Block Diagram from a Signal-Processing Point of View



MATeX

Aufgaben- und Lösungsgenerator für Höhere Mathematik in MATLAB-Umgebung mit LaTeX-Ausgabe

Andreas Helfrich-Schkarbanenko¹, Kevin Rapedius¹, Vita Rutka¹, Aron Sommer²
¹ MINT-Kolleg Baden-Württemberg am Karlsruher Institut für Technologie (KIT), Adenauerring 2, 76131 Karlsruhe
² Leibniz Universität Hannover, Institut für Informationsverarbeitung, Appelstr. 9A, 30167 Hannover

MATLAB EXPO 2016, May 10, 2016, Munich, GERMANY



Zusammenfassung

Diese Ausarbeitung richtet sich primär an Mathematik-Dozenten an Hochschulen, die für die Übungsblätterstellung LaTeX einsetzen und sich in einer Programmiersprache, insbesondere MATLAB auskennen. Wir zeigen, wie man MATLAB einsetzen kann, um mit wenigen Vorgaben für ein ausgewähltes mathematisches Thema eine vollständige Aufgabenstellung sowie Lösung samt Abbildungen in LaTeX zu generieren. Wir setzen insbesondere die Symbolic Math Toolbox von MATLAB zum automatischen Lösen von Aufgaben ein. Darüber hinaus setzen wir Systembefehle zum Kompilieren von LaTeX-Dateien und zum anschließenden Betrachten in einem PDF-Reader. Die Leistungsfähigkeit des Vorgehens wird an mehreren Themen der Höheren Mathematik am Karlsruher Institut für Technologie (KIT) vorgeführt. Mit dem vorgestellten Aufgabengenerator reduziert sich der Aufwand pro Aufgabe von 1-2 Stunden auf 1-2 Minuten. Das Ziel ist eine Abdeckung sämtlicher Standardthemen, für die sich eine generische Aufgabenstellung anbietet.

1. Hauptprogramm

```

...
ChapNr = 3; AufgNr = 1;
ChapName = ['Chapter-', num2str(ChapNr)];
AufgName = ['Ag_Taylor_', num2str(AufgNr), '_', num2str(AufgNr)];
fID2 = fopen([ChapName, '.tex'], 'r+');
switch ChapNr
    %...
    case 2
        GeneratorChap2(AufgName, AufgNr)
    %...
end
append_content(fID2, ['\input{', AufgName, '.tex'} \clearpage \n ']);
system(['pdflatex.exe ', Aufgabengenerator.tex']);
winopen('Aufgabengenerator.pdf');

```

2. Generator-Funktion

```

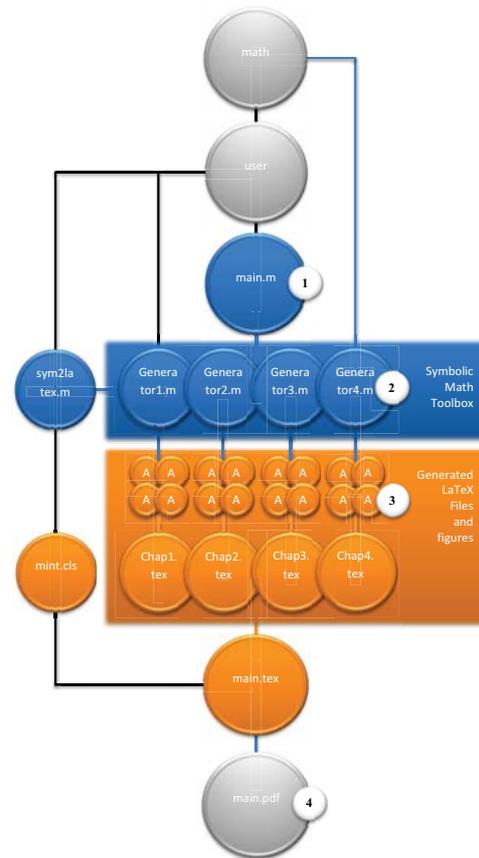
function GeneratorChap2(AufgName, AufgNr)
fID = fopen([AufgName, '.tex'], 'wt');
syms x; syms y;
switch AufgNr
    %...
    case 2
        f(x,y) = (x+y)^2-2*x + 3*y +7; x0=1; y0=-2; delta=2;
    %...
end
xl=x0-delta; xr=x0+delta; yu=y0-delta; yo=y0+delta;
append_content(fID, ['\begin{Maufgabe}{Satz} \über ... \end{Maufgabe}
...
\item Prüfen Sie nach, ob ...
\end{Maufgabe}
...
\includegraphics{', char(['Abb_zur_', AufgName, '.png']), '']);
append_content(fID, ['\ifLsg\Loesung\n '...
\begin{Maufgabe}
\item ');
dfdx = diff(f,x);
dfdy = diff(f,y);
if f(x0,y0)~=0
    Text=['Da \frac{\partial f}{\partial x}(x_0,y_0) = ...
num2str(eval(dfdy(x0,y0))), '\neq 0, ist ... \n'];
else
    Text=['Der Satz f'ur ... Funktionen ist in '...
$(x_0,y_0)^T = \left( \text{num2str}(x_0), \text{num2str}(y_0) \right)^T$ ...
\right)^T nicht anwendbar, da ... \n'];
end
else
    Text=['Der vorgegebene Punkt $(x_0,y_0)$ ... nicht ... \n'];
end
append_content(fID, Text); clear Text;

GeneratePlot(xl, xr, yu, yo, f, AufgName)

%...
append_content(fID, ['\item Die Jacobi-Matrix ist:\n '...
$$\frac{\partial f}{\partial x}(x,y) = \left( \frac{\partial f}{\partial x}(x,y), \frac{\partial f}{\partial y}(x,y) \right) = (g^T(x,y), h^T(x,y))$
\left( \text{sym2latex}(char(dfdx), 0), \text{sym2latex}(char(dfdy), 0) \right)^T$
Die ... $x_0 = \text{num2str}(x_0)$, ... \n'];
\begin{align}
g^T(x_0) &= \left( \frac{\partial f}{\partial x}(x_0, g(x_0)), \frac{\partial f}{\partial y}(x_0, g(x_0)) \right) \\
&= \left( \text{sym2latex}(char(dfdx), 1), \text{sym2latex}(char(dfdy), 1) \right)^T \cdot \left( \text{eval}(dfdx(x_0, y_0)), \text{eval}(dfdy(x_0, y_0)) \right)^T \\
&= \left( \text{num2str}(double(-1/eval(dfdy(x_0, y_0))) * dfdx(x_0, y_0)), \dots \right)^T
\end{align}
\end{Maufgabe}
]);
append_content(fID, ['\else\relax\fi \end{Maufgabe}']);

```

MATeX-Struktur



3. Generierte LaTeX-Datei

```

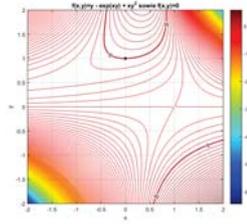
\begin{Maufgabe}{Satz} \über implizit definierte Funktionen)
Eine Kurve in $\mathbb{R}^2$ ist durch die Gleichung
$\$f(x,y) = -\exp(xy) + xy^2=0\$ implizit gegeben, siehe Abbildung.
\begin{enumerate}
\item Prüfen Sie nach, ob in einer Umgebung von
$(x_0,y_0)^T = (0,1)^T$ die implizit definierte Kurve
durch eine stetig differenzierbare Funktion $g(x)$ darstellbar
ist, d.h. $\$f(x,y)=0 \Leftrightarrow y=g(x)\$.
\item Bestimmen Sie gegebenenfalls dann $g'(x_0)$ in Punkt
$(x_0,y_0)^T = (0,1)^T$.
\end{enumerate}
\begin{center}
\includegraphics[width=0.75\linewidth]{Bilder/
Abbildung_zur_Aufgabe-2-4.png}
\end{center}
\ifLsg\Loesung
\begin{enumerate}
\item Der vorgegebene Punkt $(x_0,y_0)$ genügt der Bedingung
$\$f(x_0,y_0)=0$, d.h. der Punkt liegt auf der implizit
definierten Kurve. Da $\frac{\partial f}{\partial y}(x_0,y_0) = 1 \neq 0$, ist der Satz über die implizit
definierte Funktionen anwendbar.
\item Die Jacobi-Matrix ist:
$$\frac{\partial f}{\partial x}(x,y) = \left( \frac{\partial f}{\partial x}(x,y), \frac{\partial f}{\partial y}(x,y) \right) = (g^T(x,y), h^T(x,y))$
$= \left( -\frac{\partial}{\partial x}(\exp(xy) - xy^2), \frac{\partial}{\partial y}(\exp(xy) - xy^2) \right)$
$= \left( -y \exp(xy) + y^2, \exp(xy) + 2xy - x \exp(xy) + 1 \right)$
Die gesuchte Ableitung an der Stelle $(x_0,y_0)^T$ mit
$g(x_0) = y_0$ ist somit:
\begin{align}
g'(x_0) &= \left( \frac{\partial f}{\partial x}(x_0, g(x_0)), \frac{\partial f}{\partial y}(x_0, g(x_0)) \right) \\
&= \left( \text{sym2latex}(char(dfdx), 0), \text{sym2latex}(char(dfdy), 0) \right)^T \cdot \left( \text{eval}(dfdx(x_0, y_0)), \text{eval}(dfdy(x_0, y_0)) \right)^T \\
&= \left( \text{num2str}(double(-1/eval(dfdy(x_0, y_0))) * dfdx(x_0, y_0)), \dots \right)^T
\end{align}
\end{enumerate}
\else\relax\fi
\end{Maufgabe}

```

4. Ergebnis als PDF

Aufgabe 012 (Satz über implizit definierte Funktionen)
Eine Kurve in \mathbb{R}^2 ist durch die Gleichung $f(x,y) = y - \exp(xy) + xy^2 = 0$ implizit gegeben, siehe Abbildung.

- Prüfen Sie nach, ob in einer Umgebung von $(x_0, y_0)^T = (0, 1)^T$ die implizit definierte Kurve durch eine stetig differenzierbare Funktion $g(x)$ darstellbar ist, d.h. $f(x, y) = 0 \Leftrightarrow y = g(x)$.
- Bestimmen Sie gegebenenfalls dann $g'(x_0)$ im Punkt $(x_0, y_0)^T = (0, 1)^T$.



- Lösung:**
- Der vorgegebene Punkt (x_0, y_0) genügt der Bedingung $f(x_0, y_0) = 0$, d.h. der Punkt liegt auf der implizit definierten Kurve. Da $\frac{\partial f}{\partial y}(x_0, y_0) = 1 \neq 0$, ist der Satz über die implizit definierte Funktionen anwendbar.
 - Die Jacobi-Matrix ist:
$$\frac{\partial f}{\partial (x,y)} = \left(\frac{\partial f}{\partial x}(x,y), \frac{\partial f}{\partial y}(x,y) \right) = (g^T(x,y), h^T(x,y)) = (g^T(x_0, y_0), h^T(x_0, y_0))$$
Die gesuchte Ableitung an der Stelle $x_0 = 0$ mit $g(x_0) = y_0$ ist somit:
$$g'(x_0) = \left(\frac{\partial f}{\partial x}(x_0, g(x_0)), \frac{\partial f}{\partial y}(x_0, g(x_0)) \right)^T = \left(\text{eval}(dfdx(x_0, y_0)), \text{eval}(dfdy(x_0, y_0)) \right)^T \cdot \left(\text{num2str}(double(-1/eval(dfdy(x_0, y_0))) * dfdx(x_0, y_0)), \dots \right)^T = 0.$$

Literatur

- S. Feiler, Dokumentation zu mkbwKurseAufgaben.cls, Version 3.1, 2015
- MathWorks, Symbolic Math Toolbox™ Users Guide R2015b, 2015.
- N. J. Higham, Handbook of Writing for the Mathematical Sciences, 2nd Edition, SIAM, 1998

Danksagung

Das MINT-Kolleg Baden-Württemberg wird durch das Ministerium für Wissenschaft, Forschung und Kunst Baden-Württemberg (MWK) aus dem Programm Studienmodelle individueller Geschwindigkeit gefördert sowie im Rahmen des Bund-Länder-Programms Bessere Studienbedingungen und mehr Qualität in der Lehre (Qualitätspakt Lehre) des Bundesministerium für Bildung und Forschung (BMBF) als Verbundprojekt (Förderkennzeichen: 01PL11018A) mit der Universität Stuttgart. Die Autoren danken den Förderern MWK und BMBF für die finanzielle Unterstützung.



Lösung der Problemstellung



Abb. 1: Seilbahn zur Festung Masada (Israel)

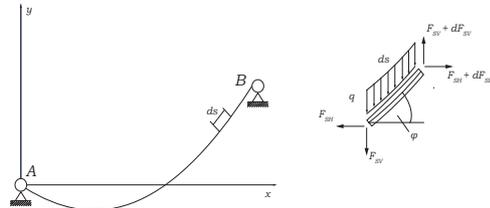


Abb. 2: Koordinatensystem und Kräfte am Seilelement ds



Abb. 3: Tibetische Hängebrücke oberhalb von Monte Carasso im Tessin

DGL: $\Rightarrow y'' = \frac{q \cdot \sqrt{1+y'^2}}{F_{SH}}$

Lösung: $\Rightarrow y = F_{SH} \cdot \cosh\left(\frac{x}{F_{SH}} + C_1\right) + C_2$

Sämtliche Größen durch charakteristische physikalische Größen dimensionslos machen!

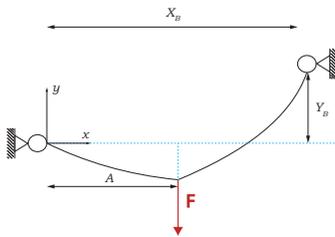
Randbedingungen: $C_2 = -\frac{F_{SH}}{q} \cosh(C_1)$, $y_B = \frac{F_{SH}}{q} \left[\cosh\left(\frac{q}{F_{SH}} x_B + C_1\right) - \cosh(C_1) \right]$, $L = \int_{x_A}^{x_B} \cosh\left(\frac{q}{F_{SH}} x + C_1\right) dx$

Auswerten und Umstellungen, sowie Eliminieren von C_1 und C_2 führt auf ein nichtlineares algebraisches GS, wobei bei bekanntem S_H die Konstanten einfach berechnet werden können:

$\left[2 \cdot S_H \cdot \sinh\left(\frac{X_B}{2 \cdot S_H}\right) \right]^2 + Y_B^2 - 1 = 0$, $K_1 = \frac{\text{arctanh}(Y_B) - \frac{X_B}{2 \cdot S_H}}{K_1 \text{ aus } S_H}$, $K_2 = -\frac{S_H \cosh(K_1)}{K_2 \text{ aus } K_1}$

\Rightarrow Nutzen von *fzero* und/oder eigenständiges Programmieren des Newton'schen Verfahrens zur Nullstellenbestimmung.

Erweiterung der Problemstellung



Es ergeben sich nun zwei Seilkurven, die geometrisch gekoppelt sind; wegen der Kräfteleitung entsteht ein Knick und über das Kräftegleichgewicht ein weiterer Zusammenhang. Es ergeben sich somit nach Normierung und Umstellung folgende Bestimmungsgleichungen für die unbekanntenen Größen:

Abb. 4: Erweiterung des Problems durch Aufbringen einer Einzellast („Gondel“)

\Rightarrow Lösung mit *fsolve*

fsolve löst das nichtlineare Gleichungssystem welches sich durch die 6 Funktionen ergibt.

- I $0 = S_H \cosh(K_1) + K_2$ Lager links
- II $0 = S_H \cosh\left(\frac{X_B}{S_H} + K_3\right) + K_4 - Y_B$ Lager rechts
- III $0 = S_H \cosh\left(\frac{A}{S_H} + K_1\right) + K_2 - S_H \cosh\left(\frac{A}{S_H} + K_3\right) - K_4$ bei $x/L=A$ stetig, aber Knick
- IV $0 = S_H \cosh\left(\frac{A}{S_H} + K_1\right) + \frac{F}{qL} - S_H \sinh\left(\frac{A}{S_H} + K_3\right)$ Vertikales Kräftegleichgewicht
- V $0 = S_H \cdot \left[\sinh\left(\frac{A}{S_H} + K_1\right) - \sinh(K_1) \right] - \beta$ Länge des Seils von linken Lager bis Kräfteleitung
- VI $0 = S_H \cdot \left[\sinh\left(\frac{X_B}{S_H} + K_3\right) - \sinh\left(\frac{A}{S_H} + K_3\right) \right] - 1 + \beta$ Länge des Seils von Kräfteleitung bis rechtes Lager

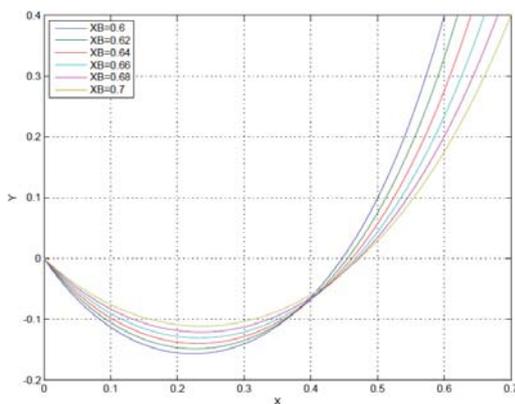


Abb. 5: Kettenlinien ohne Last

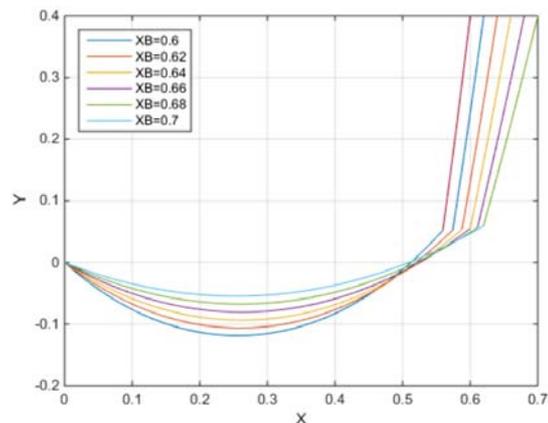


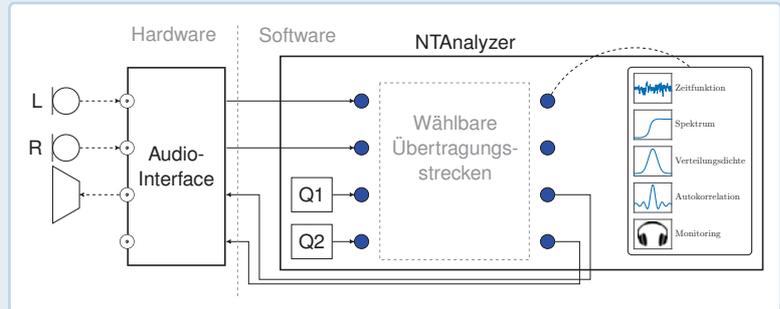
Abb. 6: Kettenlinien mit Einzellast

Max Bläser, Christian Rohlfing

Institut für Nachrichtentechnik, RWTH Aachen University

Übersicht

- Software „NTAnalyzer“ zur Unterstützung des Praktikumsversuchs „Ergodische Prozesse“ im 5. Semester des Bachelorstudiengangs „Elektrotechnik, Informationstechnik und Technische Informatik“ an der RWTH Aachen University
- Grundlagen statistischer Signalbeschreibung
 - Veranschaulichung der Zusammenhänge zwischen Zeit-, Korrelations- und Verteilungsfunktionen
 - Messung von Übertragungsfunktionen (un-)bekannter Systeme
- Echtzeitfähige MATLAB®-Software zur Signalanalyse
 - Darstellung und Wiedergabe (stochastischer) Prozesse
 - Unterstützung externer Signalquellen
 - Filterung der Messdaten in Echtzeit

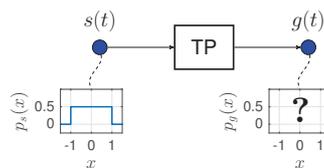


Einsatz in der Lehre

- Selbstständige Versuchsdurchführung in deutscher und englischer Sprache
- Vorhandene Versuche:
 - Statistische Charakterisierung ergodischer Prozesse durch Messfunktionen (Verteilungsdichte, Korrelation...)
 - Messung von Übertragungsfunktionen unbekannter Systeme (z.B. FIR- und IIR-Filter)
 - Messung der Schallgeschwindigkeit in Luft (zwei Mikrofone im festen Abstand)
- Typischer Messaufbau bestehend aus PC mit externem Audio-Interface, Signalgenerator, Mikrofonen und Lautsprecher

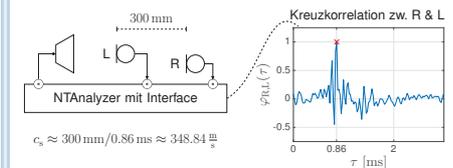
Beispiel: Transformation von Verteilungsdichten

- Wie verändert sich die Verteilungsdichte eines gleichverteilten Rauschprozesses nach Tiefpassfilterung (TP)?



Beispiel: Messung der Schallgeschwindigkeit

- Aufbau: Zwei Mikrofone mit festem Abstand und Lautsprecher (Wiedergabe von weißem Rauschen)
- Laufzeitmessung mittels Kreuzkorrelation

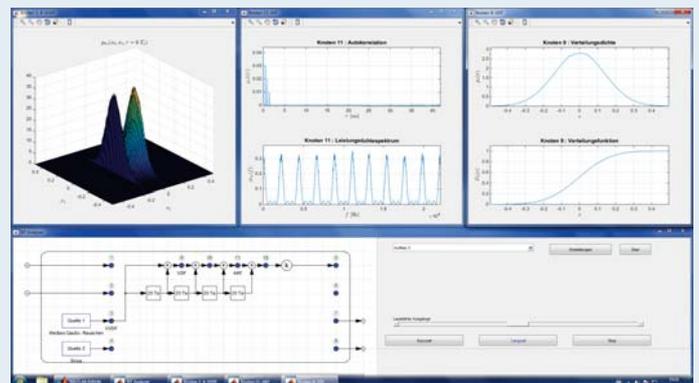


Eigenschaften der Software

- Kurz- und Langzeitmessungen in Echtzeit an Messpunkten (●):

Messfunktionen	
Autokorrelation	Leistungsdichtespektrum
Kreuzkorrelation	→ Kreuzleistungsdichtespektrum
Verteilungsfunktion	Verteilungsdichte
Zeitfunktion	Spektrum

- Signalaufnahme und -wiedergabe mit Audio-Interface
 - 2 Eingangs- und 4 Ausgangsspuren
 - Wiedergabe von internen Signalquellen und Monitoring
 - Implementierung durch `dsp.AudioRecorder` und `dsp.AudioPlayer`
- Interne Quellen Q1, Q2 (deterministische Signale und Rauschprozesse)
- Wählbare Filter und Übertragungsstrecken



Zusammenfassung

- Lehrsoftware zur Vermittlung der Grundlagen ergodischer Prozesse
- Echtzeitfähigen Analyse von Signalen und Messung von Übertragungsstrecken
 - Umfangreiche Messfunktionen und interne Signalquellen
 - Einsatz mit Mehrkanal-Audio-Interface: Anschluss von Mikrofonen, Lautsprecher, externem Signalgenerator
- Modular erweiterbar und intuitive Bedienung
- Code frei verfügbar im MATLAB® File Exchange



Teaching the Basic Principles of Game Physics by Simulating LOOP

Roland Schmitz

Studienbereich Medieninformatik, Hochschule der Medien, Stuttgart

schmitz@hdm-stuttgart.de



Introduction

The course Game Physics at Stuttgart Media University aims to convey the inner workings of a Game Physics Engine to prospective game developers, enabling them to write their own physics-based games. The basic task of a game physics engine is to simulate the physical behaviour of the so-called *Game Objects*, essentially macroscopic objects whose motion is governed by the laws of classical mechanics. More specifically, the physics engine must be able:

- to solve the equations of motions for the game objects (*Free Motion*).
- to detect possible collisions between the game objects (*Collision Detection*).
- to compute their respective behaviour after the collisions (*Collision Response*).

All these tasks need to be performed in a realistic, yet efficient way. Billiard provides a prototypic example of a game which is controlled completely by the physical behaviour of the game objects (the balls). Ball rotation and frictional effects play an important role in every real-world Billiard game and should be part of the simulation. The recent invention of LOOP [1] (Pool spelled backwards), a simple Billiard variant with four balls moving on an elliptical table with a single pocket (see Figure 1) has triggered the idea to simulate LOOP in two dimensions in MATLAB. While one would not use the current MATLAB implementation in a real game for performance reasons (see the Section on Performance below for a short discussion), it is ideally suited for illustrating the principles of Game Physics in the classroom.



Figure 1: Starting Configuration of LOOP. Left: Real Life [1]; Right: Simulated

Implementation

Free Motion

The free motion of the billiard balls is governed by two forces: The initial force on the white ball as specified by the player (see Figure 2) and the frictional force exerted by the table. The resulting simple equations of motion are solved without using the built-in solvers in MATLAB, but by the simple Euler - method instead, as is customary in game physics.

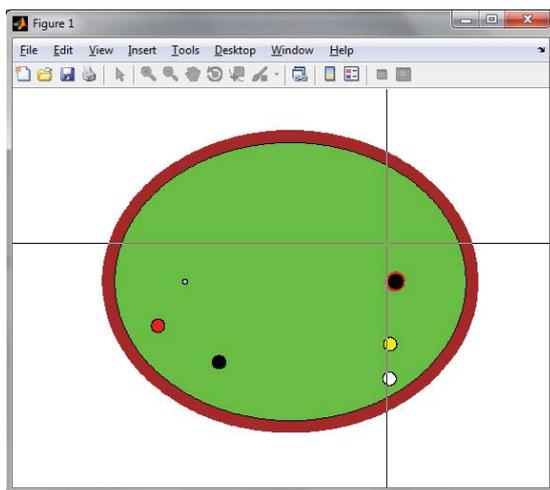


Figure 2: Screenshot of LOOP Gameplay in MATLAB. The player specifies the direction and magnitude of the force acting on the white ball using `ginput`. Note the blue dot marking the focal point opposite of the pocket. Balls passing through this point will be deflected by the table boundary towards the pocket.

Collision Detection

We have to distinguish between two types of collisions, namely Ball - Ball collisions and Ball - Table Boundary collisions. Detecting Ball - Ball collisions is a standard task in game physics, as many irregularly shaped objects are approximated by spherical boundary volumes. A Ball - Ball collision occurs if the distance of center of the balls is smaller than the sum of the radii (see Figure 3 (left)).

Detecting collisions with the table boundary are more time-consuming, as they require checking whether all points $P = (x, y)$ of the ball boundary are within the ellipse describing the table boundary, that is whether the condition

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} \leq 1$$

holds, where a and b are the half axes of the elliptic table (see Figure 3 (right)).

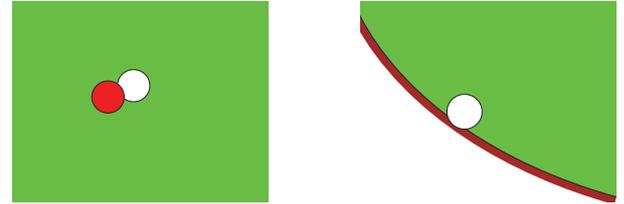


Figure 3: Detecting Collisions in LOOP: Left: Ball-Ball Collision; Right: Ball-Table Collision. Note that in both cases the objects have interpenetrated.

Collision Response

If a collision has been detected, the collision response, i.e. the behaviour immediately after the collision, needs to be computed. This is the most complex task of the simulation. For greater clarity, it is divided into three subtasks:

AnalyzeCollision

In this subtask, the actual point of collision is determined as accurately as possible. This is not as trivial as it may seem, because when a collision is detected, the colliding objects already have interpenetrated each other due to the finite size of timesteps in the simulation (see Figure 3). This interpenetration needs to be reversed in order to establish the actual point of contact CP . After that, the collision normal N can be computed, a normal vector to the surface of the colliding bodies at CP (see Figure 4).

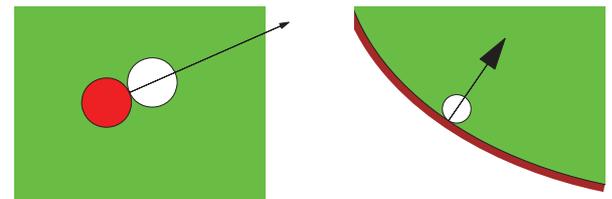


Figure 4: Analyzing Collisions. Left: Ball-Ball Collision; Right: Ball-Table Collision. The interpenetration has been reversed. The black arrow indicates the collision normal N .

CollideWithoutFriction

There are several approaches to computing the behaviour of macroscopic objects after a collision. In this simulation, we use the impulse-based framework (see [2], p. 476). Its main outcome is the magnitude of the impulse f , a large force acting over a very short time interval in the direction of the collision normal N . With the help of f , the post-collision linear and angular velocities of the colliding bodies may be computed.

CollideWithFriction

Finally, the post-collision quantities are modified to accommodate frictional effects. This entails computing a tangential impulse component f_T acting in the tangential direction T of the collision (i.e. orthogonal to N). By using f_T , it can be decided whether static or dynamic friction (see e.g. [3], p. 238) is active during the collision and the final collision responses may be determined.

Performance

The focus of the current implementation is on clarity, not on efficiency. The overall execution time on an Intel Core i7 processor with 2.80 GHz is 70 milliseconds per timestep, including rendering. Typically, however, the physics engine just updates the state of the game objects and forwards the state to the renderer. If one measures the elapsed time for updating the state only, one gets 4.2 milliseconds. The typical framerate of an interactive game is between 30 and 60 Hertz, yielding a time budget of 15 to 30 milliseconds for each timestep, of which only a few can be used for updating the physics state (see [5]). Therefore, the current, unoptimized code version is on the edge of being practical. In future work, we will try to further optimize the execution speed, e.g. by using MEX files [4].

Conclusion

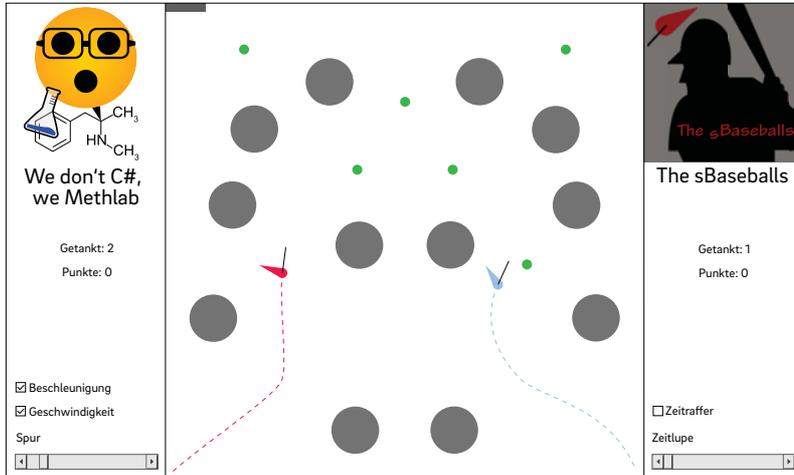
With MATLAB, the basic principles of Game Physics can be demonstrated very clearly, by simulating the game of LOOP as a showcase. The focus of the current LOOP implementation is not on speed, so there is still room for some optimization. As the simulation is in two dimensions, it is best suited to be played on a smartphone. One of the next steps will therefore be to port the MATLAB code to Android and/or iOS. Moreover, the game is currently being ported on a large multitouch-table.

References

- [1] <http://loop-the-game.com>
- [2] D. Eberly, *Game Physics*, 2nd ed. Taylor & Francis, 2010.
- [3] A. Malthé-Sorensen, *Elementary Mechanics Using Matlab*, Springer, 2015.
- [4] Mathworks®, *Introducing MEX Files*, http://de.mathworks.com/help/matlab/matlab_external/introducing-mex-files.html
- [5] M. Müller, J. Stam, D. James, & N. Thürey, *Real Time Physics: Class Notes*, in: ACM SIGGRAPH 2008 classes (p. 88), ACM, 2008.

Spaceballs

Künstliche Intelligenz in der Informatiklehre?



Prof. Dr.-Ing. Jörg J. Buchholz

<http://prof.red>

<mailto://jjbuchholz@gmail.com>

Hochschule Bremen
Abteilung Maschinenbau
Neustadtswall 30
28199 Bremen



Die Randbedingungen

- Etwa 60 Studierende
- Informatikmodul im dritten Semester
- Vierergruppen programmieren KI in MATLAB.
- Keine Vorlesung, keine Prüfung
- Turnier am Ende des Semesters
- Die Turnierrangfolge geht in die Note ein.
- Fünf Einzelgespräche mit dem Dozenten
- Wöchentliches Projekttagbuch
- Dokumentation in LaTeX
- Webseite zur Teamvorstellung

Die Lernerfolge

- Projektarbeit, Teamarbeit, Eigenarbeit
- Felder, Strukturen, Funktionen
- Schleifen, Bedingungen
- A*-Algorithmus, Traveling Salesman Problem
- Persistente Variablen, mex-files
- Kreiskollision, Tangenten an zwei Kreise
- Bahnoptimierung, Bremsweg, Orbit

Das Spiel

- Maximale Spieldauer: 60 Sekunden
- Symmetrisches Spielfeld
- Roter Spaceball ● startet links unten.
- Blauer Spaceball ● startet rechts unten.
- Banden und Minen ● dürfen nicht berührt werden.
- Wer mehr Tankstellen ● eingesammelt hat, jagt.
- Betrag der Schubkraft konstant, Richtung frei
- Alle Informationen in jedem Zeitschritt verfügbar

Die Evaluation

- Programmieren hat Spaß gemacht.
- Hohes Potential, sich MATLAB anzueignen
- Teamarbeit wird gefördert.
- Gefühl eines Großprojektes
- Sehr motivierend, klares Ziel
- Arbeitsaufwand ist sehr hoch.
- Man wird ins kalte Wasser geschmissen.
- Turnierbenotung führt zu Rivalität.
- Mindestens zwei Vorlesungen

MOOC on practical control engineering issues within the course *Dynamic Systems for Computer Scientists*

André Stollenwerk, Jan Kühn, Martin Schweigler and Stefan Kowalewski, Embedded Software, RWTH Aachen University

Motivation

- Theoretical background elaborated during course of *Dynamic Systems for Computer Scientists* (Master Studies)
- Practical experience with controller design useful in master's thesis and lab courses
- Imparts students with technical constraints and device interaction

Idea

- Workflow of control engineering on a consecutive real-world example
- Content of teaching covers
 - Plant issues like sampling rates and nonlinearities
 - Need and implementation of an anti-wind-up strategy
 - Embedded implementation of controller algorithm
 - Using MATLAB from design to plant supervision



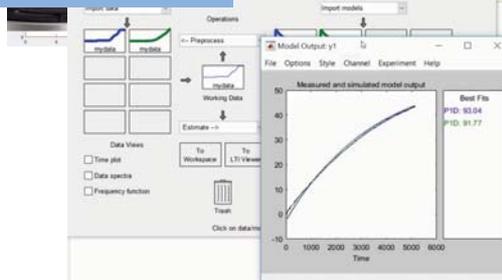
Worked example: Precise heating of mulled wine

Worked example – Closed-loop control of an electric kettle to heat up mulled wine

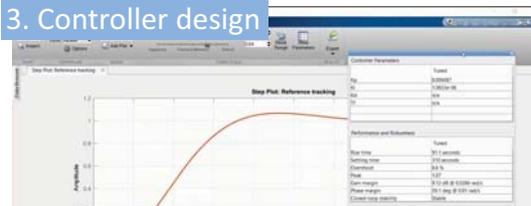
1. Taking step response



2. Plant identification



3. Controller design



4. Evaluation with disturbances



Structure of the MOOC

1. Motivation and introduction of worked example
2. Theory of system identification
3. Test signals
4. System identification of an electric kettle – measuring step responses
5. System identification – manual calculation
6. System identification using MATLAB system identification toolbox
7. Theory of controller design
8. Controller design using the PID Tuner
9. Advances in PID control
10. An outlook on stability

Presence Part as flipped Classroom exercise

- Re-record sensor readings and utilize the toolchain
- Show controller performance



<http://embedded.rwth-aachen.de>
stollenwerk@embedded.rwth-aachen.de



SCT-MAT: A MATLAB Toolbox for Supervisory Control Theory using Stateflow plant and specification models

*MOTIVATION: How to obtain a **safe-by-construction** implementation?*

Supervisory Control Theory goal:

Generate automatically correct-by-construction supervisory controllers

Academia state-of-the-art:

Nearly 30 years of research

Industry stat-of-the-art:

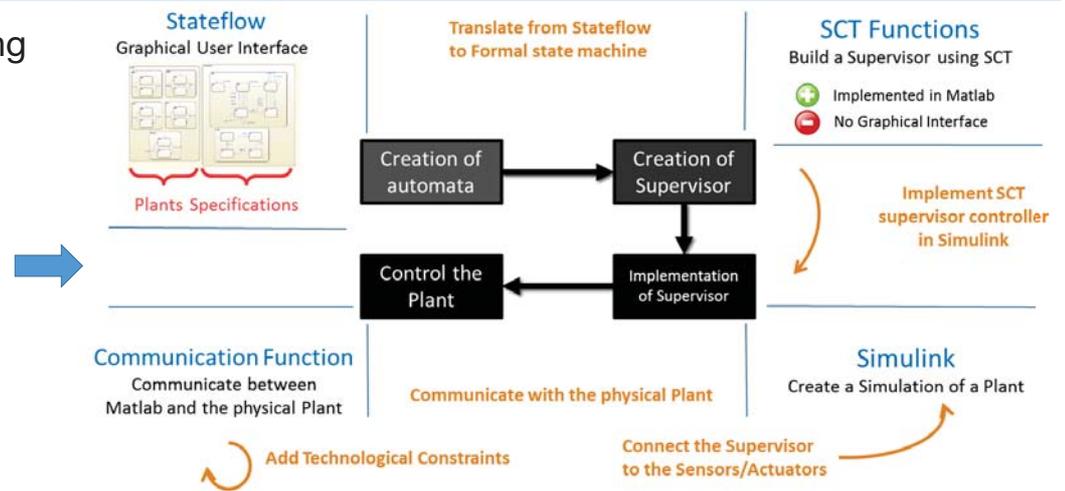
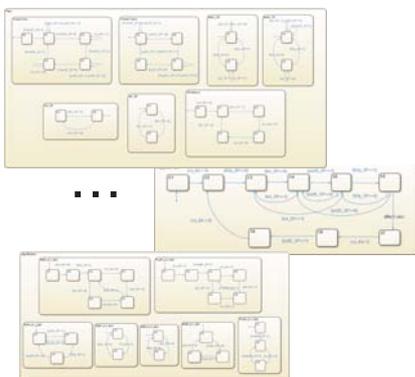
Yet, very few applications in industry

SCT-MAT tackles this industry status quo and offers:

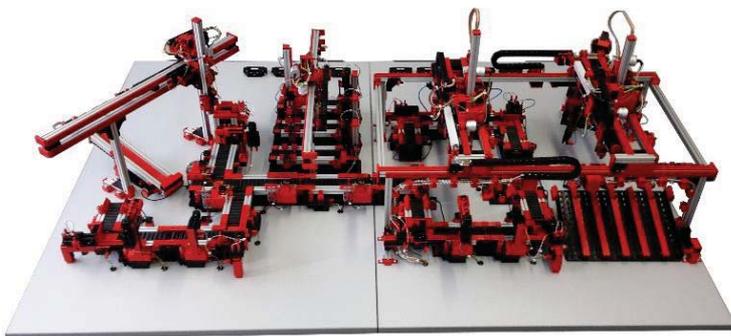
- a user-friendly graphical interface to design state machine models
- state-of-the-art algorithms to synthesize a supervisory controller
- an executable controller connected to remote input/output modules

SCT-MAT method

Plant and Specification using Stateflow models



A didactic case study



- 13 subsystems
- More than 200 sensors and actuators
- **Supervisor synthesis and code generation in a few minutes**
- **Soft-real time control over Ethernet, 50ms cycle-time**

Contact:



Prof. Dr. Julien Provost
provost@ses.mw.tum.de
089-289-16424

More information:

www.ses.mw.tum.de





DTT-MAT: A MATLAB Toolbox for Design-to-Test Approach for Testing Programmable Controllers

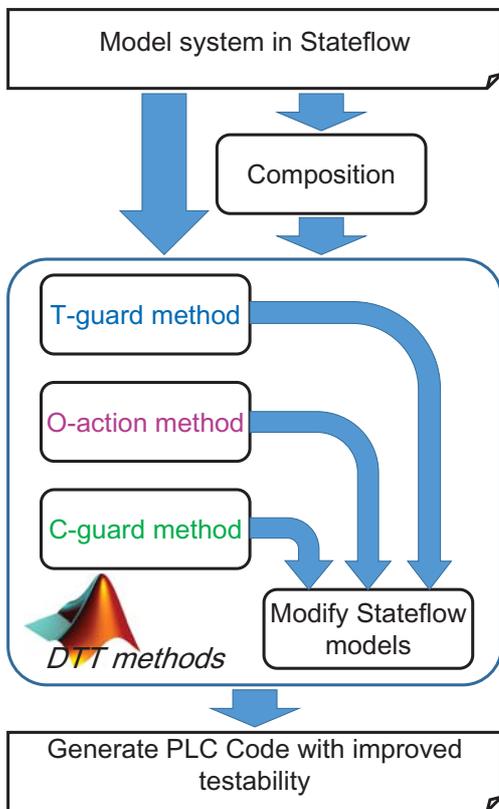
Programmable controllers



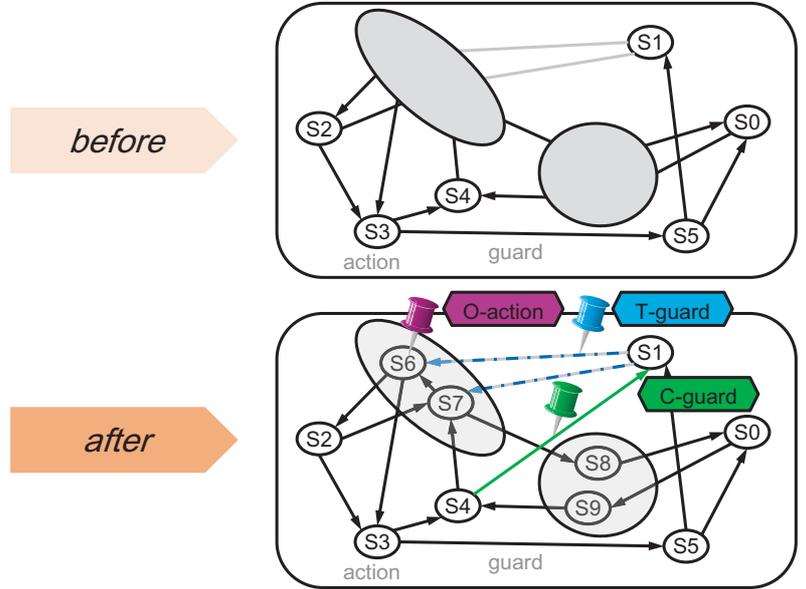
Testing issues

- SIC-testability problem
- Observability problem
- Controllability problem

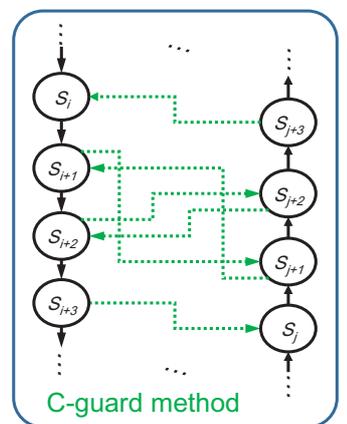
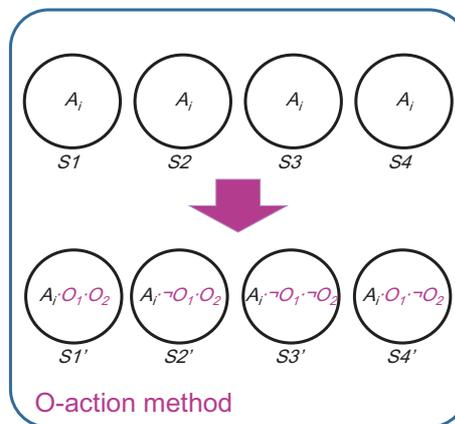
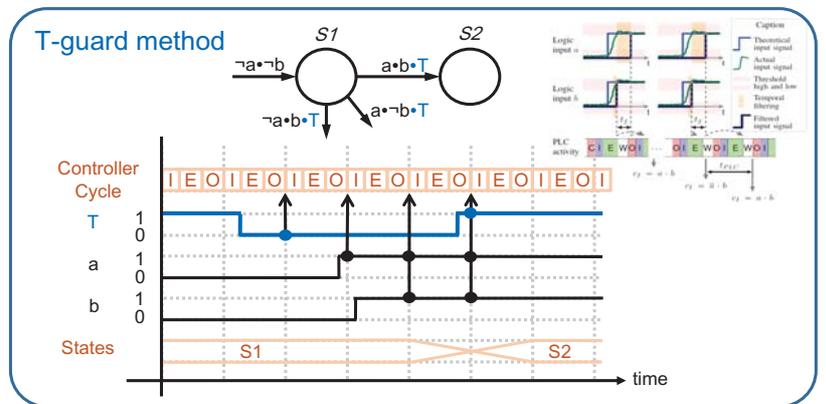
Workflow of DTT-MAT



Design-to-test (DTT) approach



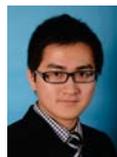
DTT methods



Contact:



Prof. Dr. Julien Provost
provost@ses.mw.tum.de
089-289-16424



Canlong Ma
ma@ses.mw.tum.de
089-289-16409

More information:

www.ses.mw.tum.de



Flexible automatisierte Erstellung von 3D Animationen mit dem MATLAB 3D World Editor am Beispiel von Schmelz- und Druckgussbetrieben

Jakob Krieg; Dominik Jeckle; Prof. Dr. Wolfgang Schlüter

Ausgangslage

Schmelz- und Druckgussbetriebe werden in MATLAB simuliert und die zeitabhängigen Ergebnisse auf verschiedenen Bildschirmen dargestellt. Auf einem Bildschirm wird eine simulationsabhängige Animation des Betriebsablaufes gezeigt.

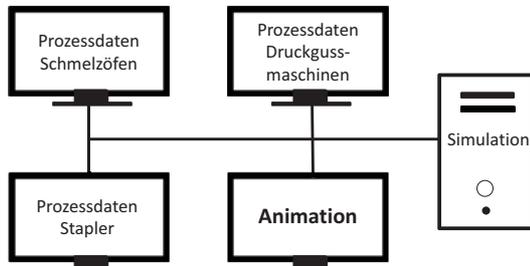


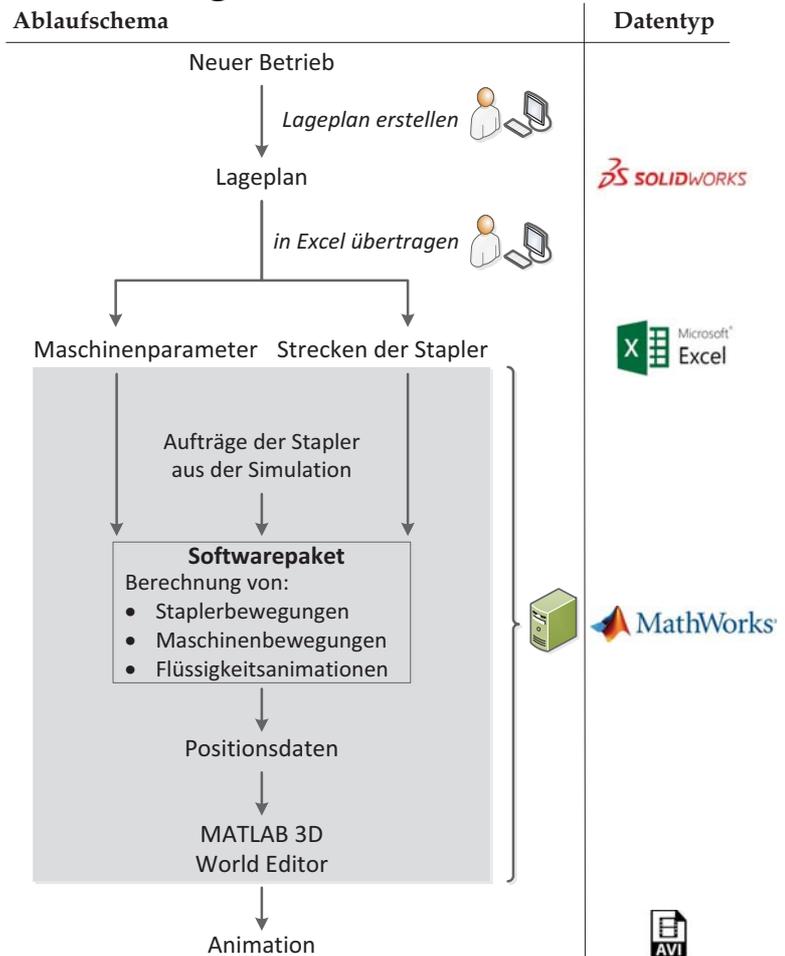
Abbildung 1: Darstellung Simulationsergebnisse

Für die Simulation eines neuen Betriebes müsste die Animation von Grund auf neu erstellt werden. Dies würde einen hohen Zeitaufwand und umfangreiche Benutzerkenntnisse beanspruchen.

Lösungsansatz

Entwicklung einer flexiblen und automatisierten Animationssoftware, mit der ein Benutzer lediglich über die Eingabe von Exceldaten beliebige Schmelz- und Druckgussbetriebe schnell und einfach animieren kann. Für die Animation wird der MATLAB 3D World Editor verwendet, der in der Simulink 3D Animation Toolbox enthalten ist. Der MATLAB 3D World Editor arbeitet mit der Virtual Reality Modeling Language (VRML).

Umsetzung



Ergebnisse



Abbildung 2: Animation

	nicht automatisiert	automatisiert
Benutzeranforderungen	Excel, MATLAB, Simulink, 3D World Editor, SolidWorks	Excel, SolidWorks
Flexibilität Betriebsparameter	nicht vorhanden	vorhanden
Berechnungsdauer	20 Minuten	2 Minuten
Erstelldauer neuer Betrieb	~ 250 Stunden	~ 15 Stunden

Abbildung 3: Vergleichswerte

Einführung

Der kaskadierte Multilevelumrichter, kurz CHB (Cascaded H-Bridge Converter), ist eine neuartige Technologie zur Anbindung stationärer Batteriespeicher an das Netz, die eine höhere Effizienz verspricht und einige Herausforderung an die Steuerelektronik darstellt. Zur Steuerung eines 17-Level-CHB werden bis zu 96 PWM-Signale benötigt, da dieser aus acht Submodulen besteht, welche jeweils 4 MOSFETs (Vollbrücke) enthalten. Die Anzahl der PWM-Signale kann auf 48 reduziert werden, indem hardwareseitig selbstinvertierende Treiber verwendet werden. Zur Regelung kann eine klassische Vektorregelung verwendet werden. In diesem Poster wird ein Ansatz vorgestellt, wie ein dreiphasiger Multilevelumrichter mit je acht Submodulen pro Phase mit Hilfe von STM32F4-Mikrocontrollern gesteuert und geregelt werden kann. Dieser Ansatz wird simulativ an einem 5-Level-Prüfstand umgesetzt und validiert. Es werden drei STM32F4-Discovery-Boards verwendet, die als Low-Cost-Mikrocontroller gelten, aber ausreichende Schnittstellen zur aufgebauten CHB-Hardware besitzen. Zur Erstellung der Software wird MATLAB® und Simulink® zusammen mit einem Blockset von Waijung verwendet. Letzteres beinhaltet bereits fertige PWM-Blöcke, die anhand von Duty Cycles (Steuerspannung des Reglerausgangs) gesteuert werden. Die Code-Erstellung und das Kompilieren finden mittels MATLAB® Code Generierung automatisch statt.

Aufbau eines 17-Level-kaskadierten Multilevelumrichters

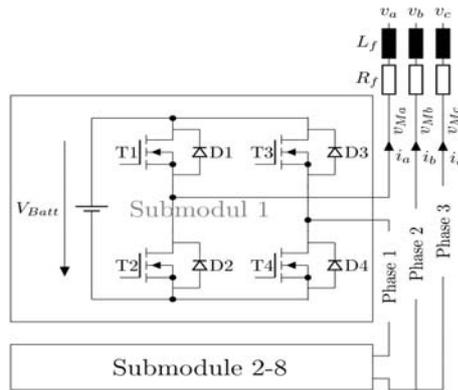


Abb. 1: Aufbau eines 17-Level-kaskadierten Multilevelumrichters

Abb. 1 zeigt den ausgewählten kaskadierten Multilevelumrichter mit jeweils acht Submodulen pro Phase. Durch physikalische Simulationen wurde das System ausgelegt und die Regler parametrisiert [1]. Jede einzelne Phase benötigt 16 PWM-Signale, die die Halbbrücken steuern. Die MOSFETs T1 und T2 werden mit dem Treiber Si8234 komplementär gesteuert, d.h. wenn der obere MOSFET angeschaltet ist, wird der untere ausgeschaltet. Damit wird ein Kurzschluss hardwareseitig vermieden.

Für die Umsetzung der Hardware werden aufgrund der großen Anzahl an PWM-Signalen drei STM32F4-Discovery-Boards benötigt. Wie diese Regler miteinander arbeiten sollen ist in Abb. 2. gezeigt.

Die wichtigsten Eigenschaften aus [2], die von diesem Mikrocontroller benötigt werden sind:

- 16 Timer (4 x Basic-Timer, 2 x Advanced-Timer, 10 x GPIO-Timer)
- 3 x Analog Digital Converter (ADC)
- 2 x Digital Analog Converter (DAC)

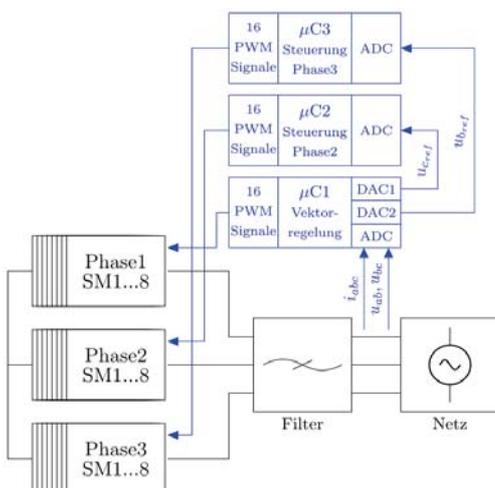


Abb. 2: Konzept zur Steuerung des Multilevelumrichters mit drei STM32F4-Mikrocontrollern

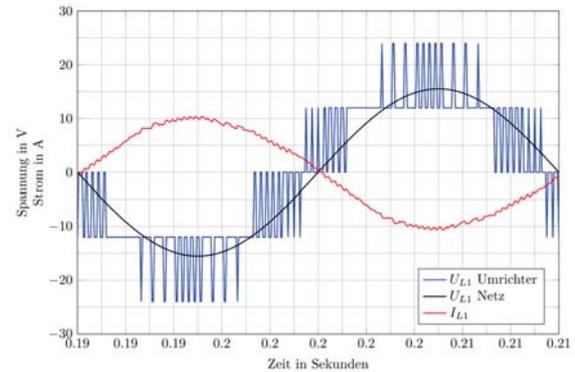


Abb. 3: Simulierte Phasenspannung, Phasenstrom und Umrichterspannung

Nach dem Simulationsergebnissen (Abb. 3) erfolgte die Umsetzung zunächst auf einem 5-Level-Prüfstand. Dabei war ein STM32F4-Discovery-Board ausreichend zur Steuerung aller 6 Submodule. Das Ergebnis der Messungen ist in Abb. 4 gezeigt.

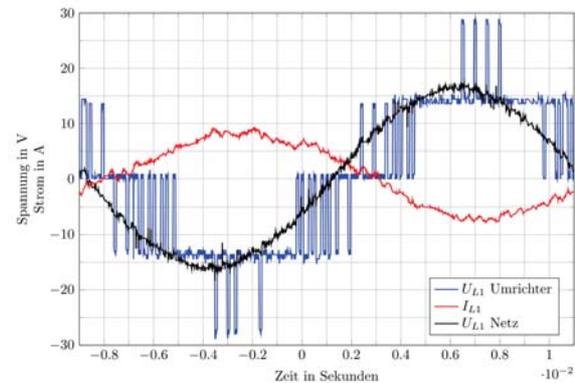


Abb. 4: Gemessene Phasenspannung, Phasenstrom und Umrichterspannung

Zusammenfassung und Ausblick

Die Umsetzung erfolgte zunächst in einen 5-Level-Prüfstand und mit einem Mikrocontroller. Nach Bereitstellung der nötigen Hardware können die restlichen Submodule und Mikrocontroller angeschlossen und verwendet werden. Die Umsetzung des Reglers erfolgt in einem Mikrocontroller, in Abb.2 als μC1 bezeichnet. Dieser nimmt zwei Leiter-Leiter-Netzspannungen und die drei Phasenströme auf und steuert die Submodule von Phase 1. An den analogen Ausgängen (DAC) des ersten Mikrocontrollers werden die Referenzspannungen für Phase 2 und 3 weitergegeben. Anhand dieser Steuerspannungen erzeugen die beiden Mikrocontroller, μC2 und μC3, die PWM-Signale für die Submodule in der zweiten und dritten Phase.

Quellen:

- [1] T. Lahlou, H.-G. Herzog, A.W. Ebentheuer, M. Herzog. Poster auf MATLAB EXPO 2015. Thema: „Multilevel-Kaskadenrichter mit einem bidirektionalen DC-DC-Wandler zur Anwendung in einem Batteriespeichersystem“. <http://www.matlabexpo.com/de/2015>
- [2] Reference manual RM0090 STM32F4. letzter Aufruf am 29.03.2016. http://www.st.com/web/en/resource/technical/document/reference_manual/DM00031020.pdf
- [3] Waijung Blockset Webseite <http://waijung.aimagin.com> letzter Aufruf am 29.03.2016.

Förderung durch

Bayerisches Staatsministerium für
Wirtschaft und Medien, Energie und Technologie



Schaltsignalerzeugung für die Ansteuerung von geschalteten Reluktanzmotoren im Überlappungsbereich

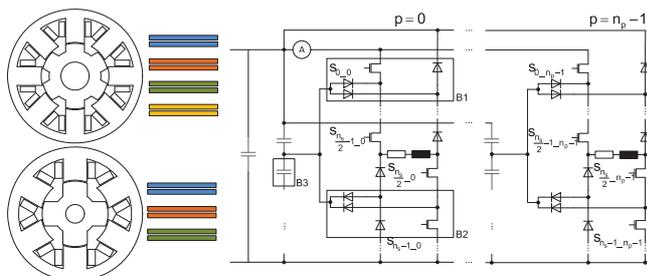
Sandro Purfürst
driveXpert GmbH
spu@drivexpert.de

Mike Eichhorn
Technische Universität Ilmenau
mike.eichhorn@tu-ilmenau.de



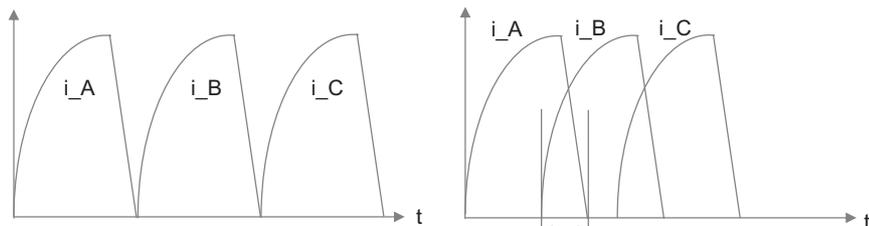
Allgemeine Betrachtung

Die verallgemeinerte asymmetrische Halbbrücke bildet die Ansteuerertopologie für geschaltete Reluktanzmotoren mit beliebig vielen Phasen und Schaltern ab. Auf diese Weise können auch Multilevelkonverter dargestellt werden.



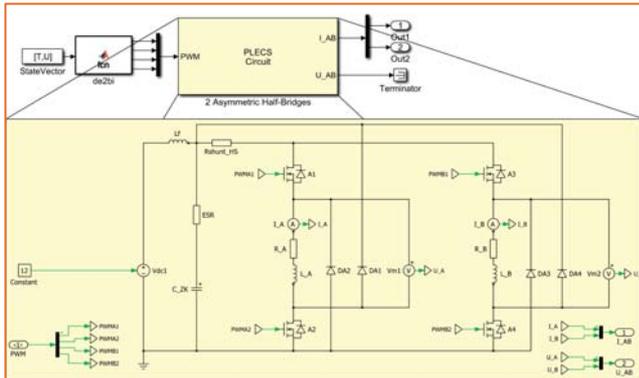
Ansteuerung

Zur Drehmomentsteigerung werden die Stromführungsintervalle der einzelnen Phasen durch die Vorverlegung des Einschaltwinkels zeitlich zur Überschneidung gebracht. Im sogenannten Überlappungsbereich entstehen somit erhöhte Anforderungen an die Messung der Ströme bei reduzierter Sensorik.



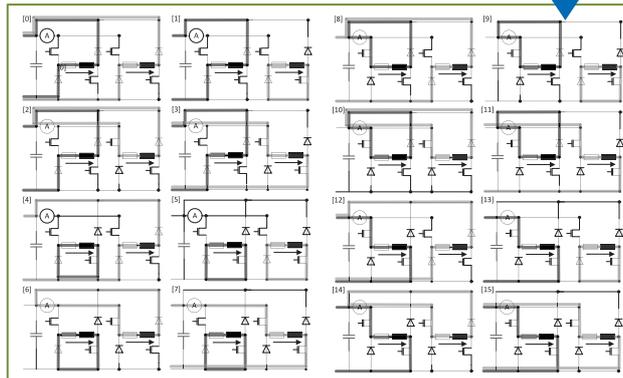
Modellierung

Mit Hilfe der PLECS Toolbox wird das elektrische Verhalten der Phasen im Überlappungsbereich nachgebildet. Somit können auch komplexere Effekte bei der Optimierung berücksichtigt werden.



Zustandsdefinition

Im Überlappungsbereich werden jeweils 2 Phasen gleichzeitig bestromt. Für den einfachen Fall mit zwei Schaltern je Phase ergeben sich somit 16 verschiedene Bestromungszustände.



Eigenschaften

Die Eigenschaften beschreiben Phasenspannung und Strommessmöglichkeiten.

Datenstruktur

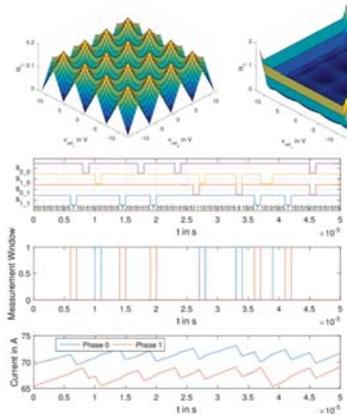
Die Zustände der Schalter werden als binäre Zahl interpretiert, deren dezimale Entsprechung als Index für den Zugriff auf die im Programm hinterlegten Eigenschaften dient.

$$V_{state} = \sum_{y=0}^{n_p-1} \sum_{x=0}^{n_p-1} S_{xy} (2^{n_p n_p - y n_p - x - 1})$$

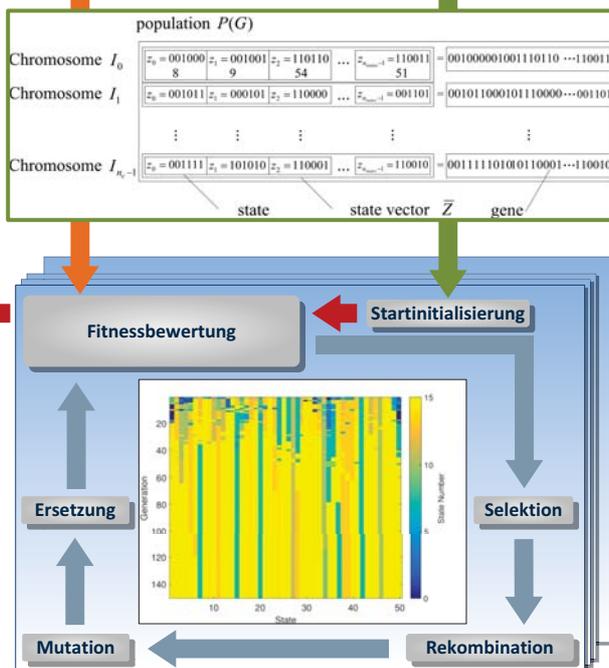


Genetische Optimierung

Die zeitliche Abfolge der Leistungshalbleiterschaltzustände als Gencode eines Lösungsindividuum (Zustandsvektor) bildet die Grundlage für die Arbeit eines GA. Dabei werden Zustandsvektoren generiert, die den gegebenen Kriterien genügen. So ergibt sich für jede geforderte Spannungs-kombination eine Schaltfolge, die mit einem minimalen Fehler von der Vorgabe für die Sollspannungen abweicht und eine Menge von Messfenstern bei minimalem Stromrippel bereitstellt.



Die genetisch optimierten Schaltmuster werden mit der Global Optimization Toolbox generiert.



Optimierungskriterien

Unter Zuhilfenahme der Zustandseigenschaften und der PLECS Toolbox werden u. a. folgende Kriterien in Gestalt einer Fitnessfunktion formuliert:

- Minimaler Fehler zwischen vorgegebenem und erreichtem Phasenspannungsmittelwert
- Minimaler Stromrippel
- Erreichen einer definierten Anzahl von Strommessfenstern

$$fit = w_{voltage} fit_{voltage} + w_{ripple_0} fit_{ripple_0} + \dots + w_{ripple_{n_p-1}} fit_{ripple_{n_p-1}} + w_{m_0} fit_{window_0} + \dots + w_{m_{n_p-1}} fit_{m_{n_p-1}}$$

Parallelisierung

Durch die Nutzung der Parallel Computing Toolbox kann die Berechnungsdauer bei 16 Kernen auf ein Zwölftel reduziert werden. Die Optimierung wird abgebrochen sobald nach einer bestimmten Zeit keine Verbesserung eintritt.



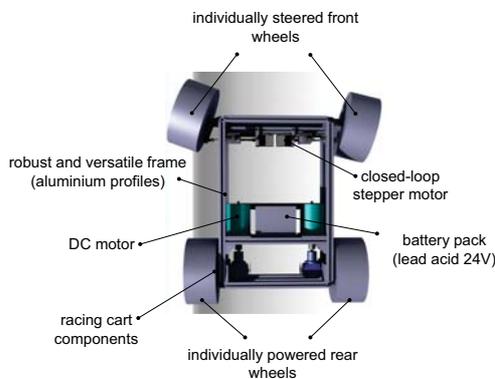
„IRT-Buggy“ – Autonomous Navigation Platform

Experimental Vehicle, Development Tool and Platform for Education

Experimental Vehicle



IRT-Buggy with High-Precision Navigation Equipment



Overall Specs:

Mechanical Setup:

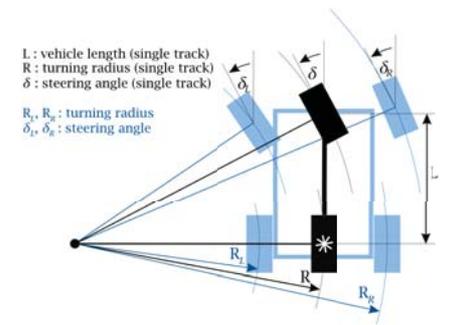
- Robust and versatile frame made from aluminium profiles
- Efficient and fast to fabricate and to adapt
- Easily extendable (size, external components)
- Racing kart components (rims, tires, axles, hubs)
- Overall weight ~60kg

Propulsion: 2 Brushed PMDC Motors

- Easy to control (torque ~ current)
- Independently driven („virtual differential“)
- Peak power: ~2x1.2kW
- max speed: ~50km/h (depending on transmission ratio)
- Acceleration: ~2.5m/s² (depending on transmission ratio)

Steering: 2 Closed-Loop Stepper Motors

- Independent steering angles for left/right side
- Integrated power electronics
- Steering radius (center point): $R_{min} = 1.4m$



Ackermann Steering: Set-Speeds for Wheels and Steering Angles are Individually Calculated for Each Wheel Based on Single-Track (Bicycle) Model

Electronics & Controls

General Concept

- Designed to be used for control engineering purposes
- Built around sensors & actuators
- Easy to model (2D, simple and known kinematics)
- Simple and full access to all sensors and actuators
- Almost „Plug&Play“ use for higher control tasks

Operational Modes

- Operational mode chosen through RC controller
- Operator can always regain full control over vehicle („Safety Pilot“)

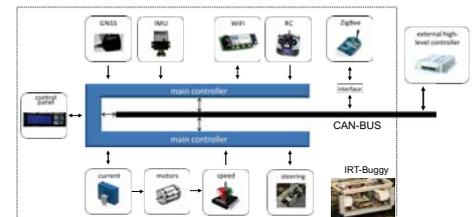


Baseline Control

- TI C2000 DSP Controller (150MHz)
- Model-based software design using MATLAB/Simulink
- Interface to sensors and actuators
- Low-level control loops (current, speed, steering)
- GNSS decoding functionality (NMEA format)
- CAN interface to higher-level controller

Sensor Equipment

- 9 DOF Inertial measurement unit (accelerometers, gyroscopes, magnetometers)
- Incremental wheel speed sensor for each wheel
- Current sensors
- GNSS Sensor
- Communication Interfaces (WiFi, ZigBee)



Development Tool:

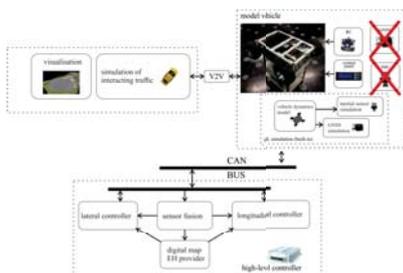
Filling Missing Link in Development Workflow

- Much lower hardware cost when compared to full-scale vehicle
- Control hardware and sensors very similar to full-scale vehicle
- Allows to achieve research results with high validity for many applications at a fraction of the cost



Special Feature: Built-In HiL Simulation

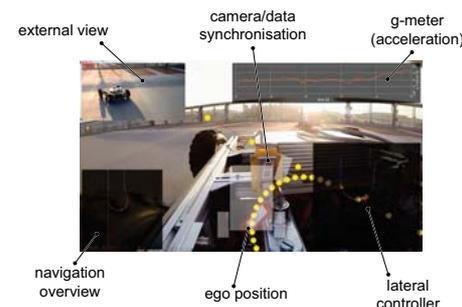
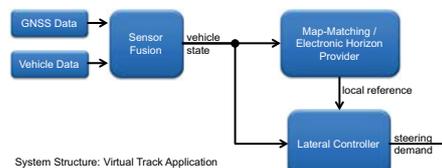
- Built-in HiL Model as integral part of baseline controller
- IRT-Buggy can „simulate itself“
- Full test of high-level controller functionality (application-dependent)
- Save debugging time on the test track by first testing in the lab („if it doesn't work in the lab, it for sure will not work on the track“)



Navigation-Based Control Systems:

Example Application: Virtual Track

- Fusion of GNSS and inertial sensor data to obtain high-speed vehicle position, heading and velocity
- Integration of map data: Software provides nearby map points
- Lateral controller takes over steering actuation, vehicle is „hooked“ onto track described in map
- Idea: Reducing the number of degrees of freedom eases up the experimental setup, e.g. to perform longitudinal control of many interconnected vehicles



Platform for Education:

Things to Teach and Learn

- Model-Based software development
- Control engineering tasks
- Use of modern control engineering hard- and software
- Getting into touch with real sensor data

Versatility and Ease of Use

- Low-power-mode allows for safe operation after very little introduction time
- Access and close control loops using professional control hardware or just a regular laptop
- Suitable for use indoors and outdoors



Good Payload



Student Competition: Lateral Control Based on Vision



Inverted Pendulum/Segway Mode

In the Making: IRT-Buggy Mk2

- Improved packaging&integration
- Lighter&faster
- Removeable battery-pack

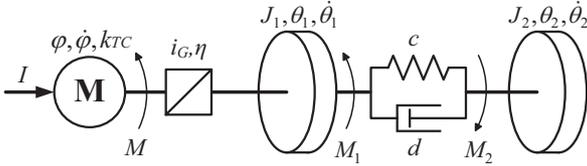


Ziele der Mastervorlesung Mehrgrößenregelsysteme (4 SWS, 5 ECTS-Credits)

- Vermittlung regelungstechnischen Fachwissens zur Synthese, Parametrierung und Implementierung von Regelungen für Mehrgrößensysteme
- Einarbeitung in fünf Fachthemen (mit jeweils vier Teilthemen) anhand von Primärliteratur sowie kooperatives Lernen in Stammgruppen, Einsatz von Blended Learning
- Vernetzung von fachlicher und überfachlicher Projekt- und Sozialkompetenz
- Applikation des Fachwissens an einem mechanischen Mehrgrößensystem in Simulation (MATLAB/Simulink) und Experiment
- Eigenverantwortliche Evaluierung des Lernfortschritts

Aufgabenstellung

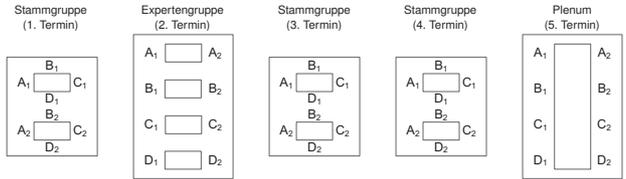
Schwingungsdämpfungsregelung eines Zweimassenschwingers als Ersatzmodell eines PKW-Antriebsstrangs



→ Regelung des abtriebsseitigen Winkels θ_2 bei Messung des antriebsseitigen Winkels θ_1

Methode Gruppenpuzzle zur Bearbeitung eines Fachthemas

Beispiel: Zwei Stammgruppen bearbeiten ein Fachthema mit vier Teilthemen ($\hat{=}$ Gruppenstärke)



A, B, C, D: Experte (gruppenintern benannt, arbeitet sich vertieft in ein Teilthema ein)
 1, 2: Nummer der Stammgruppe

Fachthemen

1 Zustandsraummodell

```

syms c d J_1 J_2 k_TC i_G

A = [0 0 1 0;
     0 0 0 1;
     -c/J_1 c/J_1 -d/J_1 d/J_1;
     c/J_2 -c/J_2 d/J_2 -d/J_2];
B = [0;
     0;
     i_G*k_TC/J_1;
     0];
C = [1 0 0 0];
D = 0;

MRS_Params;
A = eval(A);
B = eval(B);
C = eval(C);
D = eval(D);

system = ss(A,B,C,D);
pzmap(system)
    
```

2 Mehrgrößensysteme

```

% Steuerbarkeit
Ss = ctrb(system);
rank(Ss)

% Beobachtbarkeit
Sb = obsv(system);
rank(Sb)

% Eigenwerte
EW = eig(A);

% ROSENBRÖCK-Systemmatrix
syms s
P = [s*eye(4) -A B; C D];
    
```

3 Mehrgrößenreglerentwurf und -simulation

```

% Zustandsrückführung
schalter = 2;
if schalter == 1 % Polplatzierung
    p = [-13; -14; -15; -16];
    K = place(A,B,p);
elseif schalter == 2 % Optimalregler
    Q = diag([0 40 0 0]); R = 5; N = 0;
    K = lqr(system,Q,R,N);
else
    disp('schalter = 1 / 2');
end

% Vorfilter
V = -inv(C*inv(A-B*K)*B);

% LUENBERGER-Beobachter
p_B = 2*p;
L = place(A',C',p_B)';

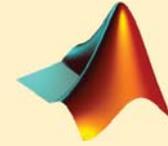
% Federsteifigkeit in Strecke
% 20% geringer als im Beobachter
c2 = 0.8*c;
A2 = [0 0 1 0;
     -c2/J_1 c2/J_1 -d/J_1 d/J_1;
     c2/J_2 -c2/J_2 d/J_2 -d/J_2];
    
```

4 Implementierung auf Prüfstand mit Simulink Real-Time

Verwendete Toolboxes

MATLAB/Simulink, Symbolic Math Toolbox, Control System Toolbox, Stateflow, Simulink Real-Time (allen Studierenden zur Verfügung gestellt über die Total Academic Headcount-Lizenz)

Ein FM/RDS-Empfänger mit MATLAB/Simulink und RTL-SDR (Software Defined Radio) – eine ideale Kombination für die Lehre



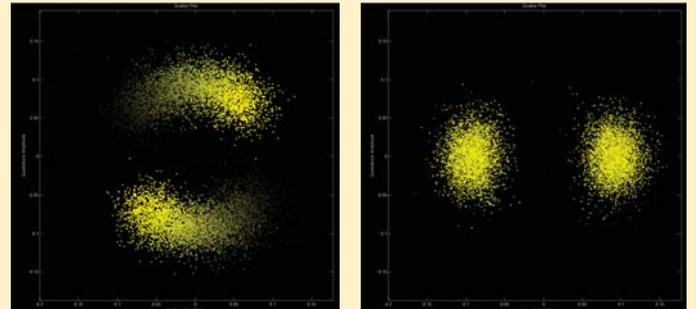
RTL-SDR

- USB-Stick + externe Antenne
- Sehr kostengünstig
- Hardware Support Package für MATLAB/Simulink verfügbar
- Erforderliche Toolboxes:
 - Communications System Toolbox
 - DSP System Toolbox
 - Signal Processing Toolbox
- Streaming der I/Q-Komponenten über USB
- Einschränkungen:
 - Nur Empfänger (kein Sender)
 - Frequenzbereich ca. 25 MHz bis 1.7 GHz
 - Maximale Bandbreite ca. 2.8 MHz
 - Limitierte Empfindlichkeit

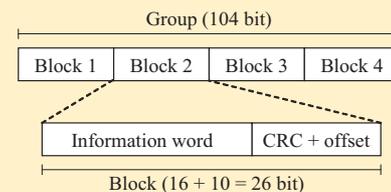


RDS-Empfänger

- BPSK-Demodulator
- Downsampling auf 1 Sample/Bit (Bitrate 1187.5 bit/s)
- Träger- und Symboltaktssynchronisation
- Block- und Gruppensynchronisation
- Decodierung der RDS-Nachrichten



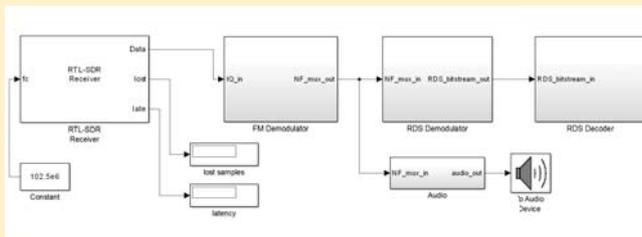
Scatter Plots vor und nach der Trägersynchronisation (10⁴ bit)



Gruppen- und Blockstruktur des RDS-Signals

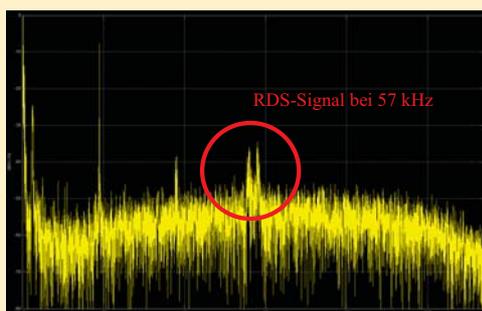
FM-Rundfunkempfänger

- Audio-Wiedergabe einer Rundfunkstation



Simulink-Modell des FM/RDS-Empfängers

- Komponenten des demodulierten Signals:
 - Audiosignal bis 15 kHz
 - Stereopiloton bei 19 kHz
 - Stereomultiplexsignal bei 38 ± 15 kHz
 - RDS (Radio Data System)-Signal bei 57 kHz, Bandbreite 4.8 kHz



Spektrum eines demodulierten FM-Rundfunksignals

Laborpraktikum mit RTL-SDR + FM/RDS-Empfänger + MATLAB/Simulink

- Sehr kostengünstig Lösung, die für mehrere Arbeitsgruppen beschafft werden kann
- Untersuchung eines analogen Modulationsverfahrens (FM)
- Untersuchung eines digitalen Modulationsverfahrens (BPSK)
- Echtzeit-Betrieb
- Aufbau des Empfängers ist genügend einfach, um in begrenzter Zeit zum Erfolg zu kommen
- Verwendung in einem Master-, aber durch entsprechende Vorgaben und Hinweise auch in einem Bachelorstudiengang möglich
- Bei erfolgreicher Implementierung unmittelbares Erfolgserlebnis für die Studierenden (z. B. Stationsname, decodierte Textnachrichten u. v. m.)



Literatur

- Stewart, R. W. et al.: *A Low-Cost Software Defined Radio Design Environment Using MATLAB, Simulink, and the RTL-SDR*. IEEE Communications Magazine, September 2015.
- Roppel, C., Perez, A. O.: *An Efficient Implementation Of An FM/RDS Software Radio*. European DSP Education and Research Conference EDERC 2010, Nice (France).
- Roppel, C.: *Grundlagen der digitalen Kommunikationstechnik*. Hanser-Verlag, 2006.

Software Radio in der Lehre mit dem HackRF: Simulink-HackRF Blockset und MATLAB Interface

Sebastian Koslowski, Friedrich K. Jondral
Communications Engineering Lab, Karlsruhe Institute of Technology
Sebastian.Koslowski@kit.edu, Friedrich.Jondral@kit.edu

1. Einleitung

Motivation: Universitäre Lehre; Labor zum Grundlagen Nachrichtentechnik

- Over-the-Air-Übertragungen als Zusatz zu Simulationen
- Unterschiede zwischen echten und simulierten Übertragungen herausarbeiten
- Begeisterung für Nachrichtentechnik

Software Radio Peripheriegeräte

- Enthält konfigurierbares analoges und digitales Front-End
- Abtastwerte werden in Echtzeit vom und zum Gerät übertragen
- Signalverarbeitung am PC in Software (z.B. MATLAB / Simulink)
- Vergleichsweise günstige Möglichkeit für viele Arbeitsplätze
- Produkte (Beispiele):
 - Ettus Universal Software Radio Peripheral (USRP): leistungsfähig, aber teuer
 - RTLSDR-Dongle: nur Empfänger, beschränkter Frequenzbereich, dafür sehr günstig
 - HackRF: Sender und Empfänger, vergleichsweise günstig

Ziel des Projekts

- Senden und Empfangen von Signalen mit günstiger Hardware
- Signalverarbeitung in MATLAB / Simulink → Interface zu Hardware wird benötigt

2. Hardware und Systemaufbau

„HackRF One“ Software Radio Peripheral by Great Scott Gadgets

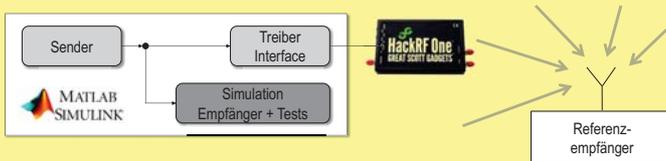
- Open Source Hardware Plattform zum „Testen und Entwickeln“ von SDR-Anwendungen
- Kosten: 280€ zzgl. Antenne



- USB 2.0 HiSpeed zur Stromversorgung und Ansteuerung von einem PC (C API)
- Half-duplex low-cost Transceiver bei einer Mittenfrequenz von 1 MHz bis 6 GHz
- Maximal 20 MS/s, 8 Bit ADC, 10 Bit DAC

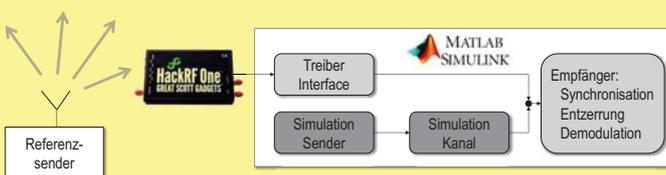
Szenario 1: Alle Senden (abwechselnd) zu einem Referenzempfänger

- Aufgaben: Sender implementieren, in Simulation testen, dann Signal aussenden.



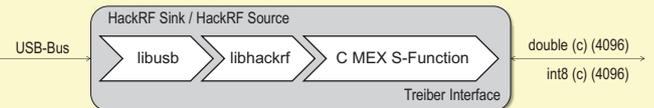
Szenario 2: Alle Empfangen und Dekodieren das Signal eines Senders (Rundfunk)

- Aufgaben: Empfänger vervollständigen, mit synthetischem Signal testen, dann empfangen



3. Simulink-HackRF Blockset

Aufbau

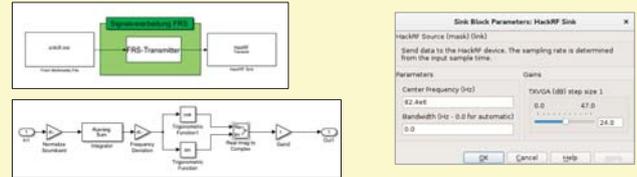


Funktion

- Initiale Konfiguration der Hardware, Veränderung der Parameter im Betrieb
- Senden/Empfangen von komplexen Basisbandsamples bei gewünschter Bandbreite/Mittenfrequenz
- Installation: freier Compiler (MinGW), getestet mit Linux und Windows

Beispiel: Sender für Sprechfunkgeräte, Family Radio Service (FRS)

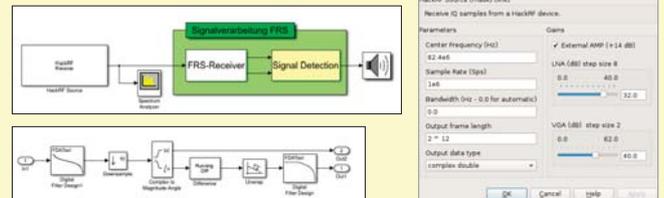
- Aufgabe: Audiosignal FM-moduliert aussenden



Beispiel: Einfacher Empfänger für FRS-Signale

- Aufgabe: Empfangenes Signal demodulieren und ausgeben

- Weitere: Rauschunterdrückung, Rufnumerkennung



4. MATLAB Interface

Funktion

- Ermöglicht unkompliziert Aufzeichnungen fester Längen mit den übergebenen Parametern
- Signalverarbeitung erfolgt offline

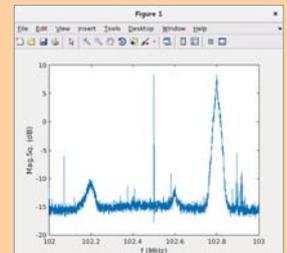
```
% Initialize hackrf device
handle=hackrf_dev(0);

bins = 4096;
N = 100;
B = 1e6;
fc = 102.5e6;
f = linspace(-B/2, B/2, bins + 1); f = f(1:bins);

% data = hackrf_dev(handle, 'RX', 1e6, 1e6, fc, [25,24,14], N * bins);

data = reshape(data, N, bins);
S = mean(abs(fshift(fft(data, bins, 2)))^2 / 4096);
plot(fc + f) / B, 10 * log10(S));
xlabel('f (MHz)'); ylabel('Mag_Sq, (dB)');

% close device handle
hackrf_dev(handle);
```



- (contributed by Jiang Wei)

5. Zusammenfassung

- „Yet Another SDR interface“
- Hands-On Software Radio in der Lehre mit MATLAB Simulink
- Versuchsaufbau mit kostengünstiger Hardware für „viele“ Arbeitsplätze
- Projekt Homepage: <https://github.com/kit-cell/simulink-hackrf>

