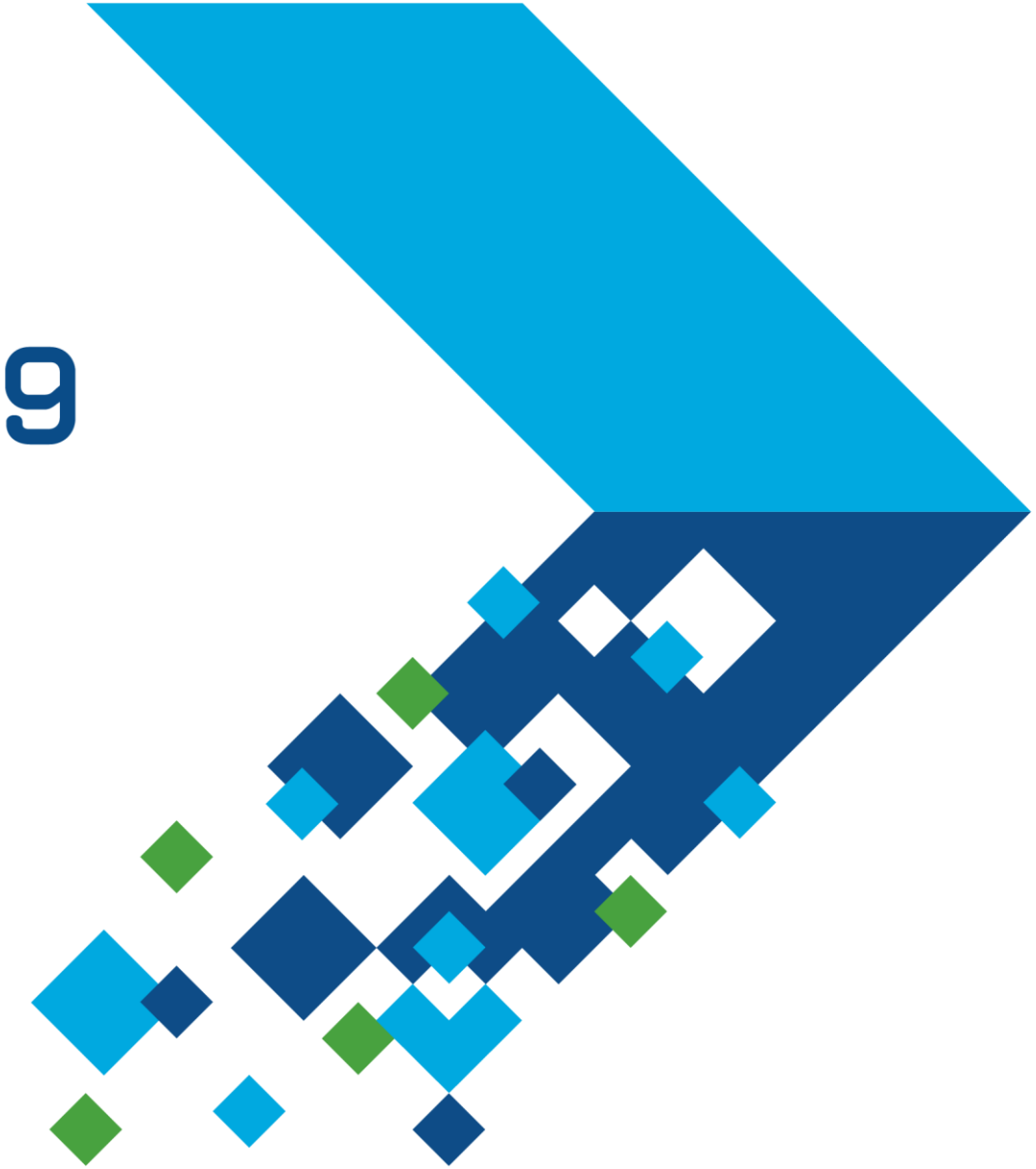
MATLAB EXPO 2019

Postersession



Poster 2019

1 - Mathematik-Universum interaktiv erkunden - Generieren von Universal 3D–Abbildungen für die Lehre

2 - Vorlesungsunterstützung durch MATLAB im Bereich Technische Mechanik

3 - Digitaler Rückenwind für Mathematikveranstaltungen - Automatisches Generieren, Durchführen und Auswerten von Mathematiktests in Moodle

4 - Das Elektronik Praktikum in der eigenen Bude - Digitaltechnik mit Simulink als Telematik Versuch

5 - Individualisierte Begabtenförderung mit MATLAB & Simulink - Gezielte MINT-Förderung an der Schnittstelle Schule-Universität

6 - Mobiler Temperaturregelungsversuch

7 - Position Control of Brushless DC-Motors in a Low-Cost Laser Deflection Device

8 - MOBATSim: MOdel-Based Autonomous Traffic Simulation Framework

9 - Comparison of 4 Different Routing Algorithms in MATLAB

10 - Helikopter - Flugsimulation

11 - Hardware in the Loop Experimente für die Multicopter Entwicklung

12 - The Hohmann-Transfer

13 - Evaluation of three different theories about the future of our universe

14 - Simulation des Falls in ein schwarze Loch anhand der Schwarzschild-Metrik

15 - BACnet stack integration with BACnet4Simulink

16 - Mbed Target and IOT – A ThingSpeak Simulink® Blockset

17 - Indoor Position Data Aggregation and Indicator Extraction for Production and Logistic Issues with MATLAB®

18 - A Dynamic Sensor Fault Detection and Identification System in IoT using MATLAB and Simulink

19 - AI based virtual sensor & durability engineering

20 - Neural Networks

21 - Facial Recognition with Convolutional Neural Networks

22 - Image Processing in Electrochemistry - Measuring ionomer layers in a fuel cell electrode

23 - Dynamo: A MATLAB based scientific library for cryo electron microscopy

24 - High Speed Thermography Mapping

25 - Plattformunabhängige NC-Kernerweiterung mit Matlab\Simulink

26 - Automatic LON-Bus Model Generation using SimEvents

27 - MatCL - Eine einfach anzuwendende OpenCL Toolbox für MATLAB

28 - Problems of electronics engineering

29 - Analog co-simulation of a RLC- Circuit with MATLAB/Simulink and Cadence ADE

30 - Price Setting in Monopolistic and Oligopolistic Markets

31 - Rot-Schwarz-Baum - Ein balancierter binärer Suchbaum

32 - Numerical Linear Algebra in Real World Applications

33 - Nutzung des ADM-Modells zur Optimierung des Methanflusses in Biogasanlagen

34 - Material- und Energieflusssimulation in Aluminium-Druckgussbetrieben

35 - Development of optimal fast-Charging Strategies using a validated battery model

36 - WiFO: Multi-dimensional Wind Farm Layout Optimization

37 - Smart Grid modeling in MATLAB-Simulink

Präsentationsmodus Postersession

- Vorstellung der Poster in Kurzvorträgen
- 60 Sekunden pro Sprecher
- Fliegender Wechsel

Please vote - Best Poster Award



or

http://tinyurl.com/y5d98n6p
(https://www.surveymonkey.de/r/72TQFKV)

MATLAB EXPO 2019

Hochschule Esslingen
University of Applied Sciences

MATHEMATIK-UNIVERSUM INTERAKTIV ERKUNDEN

GENERIEREN VON UNIVERSAL 3D-ABBILDUNGEN FÜR DIE LEHRE

Dipl. Inf. Achim Eichhorn, Prof. Dr. Andreas Helfrich-Schkarbanenko, Fakultät Grundlagen

Visuelle Medien sind wichtige Instrumente beim Vermitteln mathematischer Inhalte. Wir zeigen auf, wie man für Lehrunterlagen das Universal 3D – Datenformat heranziehen kann, wodurch sich dreidimensionale Abbildungen direkt im PDF-Dokument interaktiv erkunden lassen. Die Erstellung von Abbildungen sowie ihre Einbettung in PDF-Unterlagen erfolgt durch eigene MATLAB-Skripte.

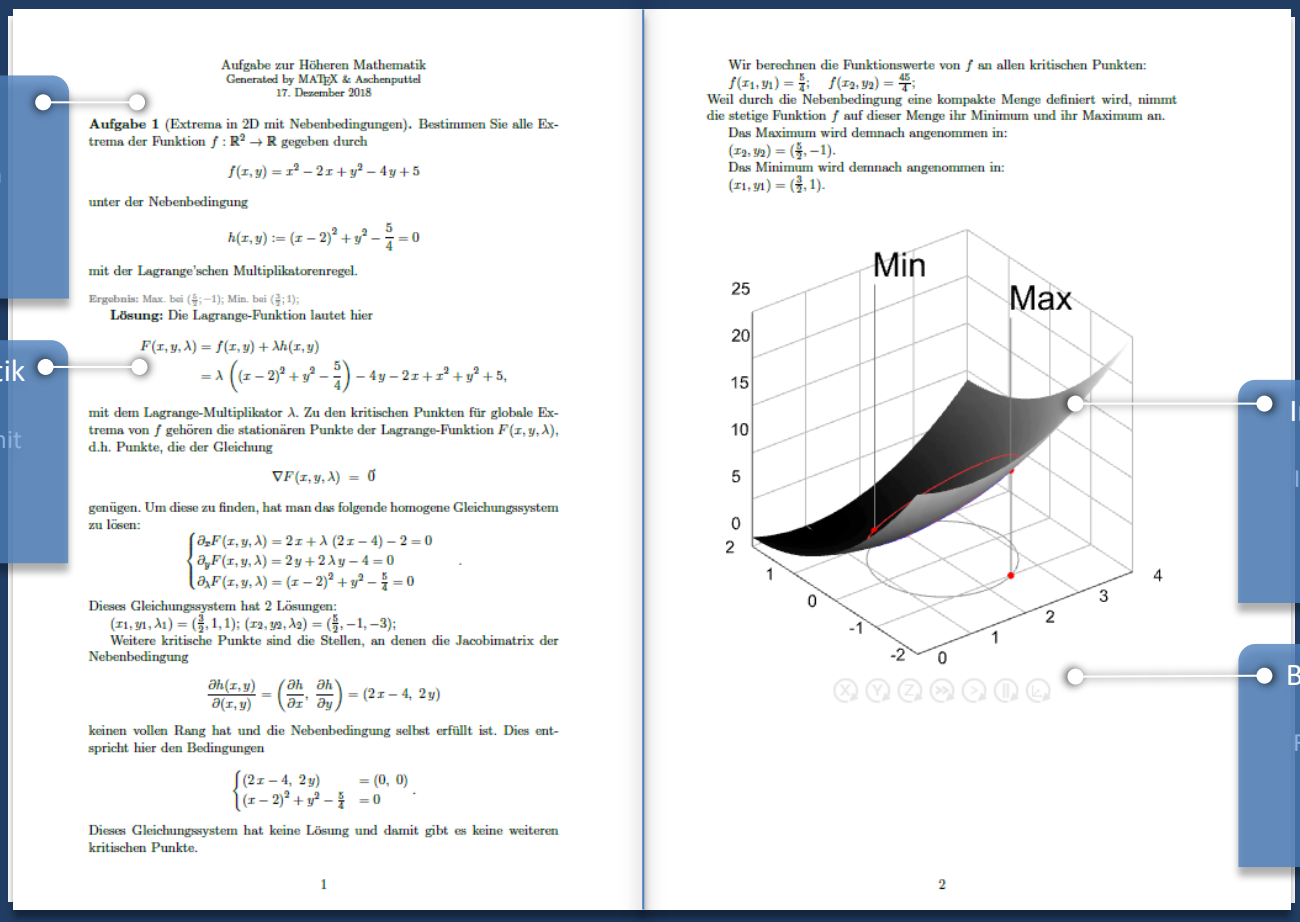
Ferner wurde die frei verfügbare, in der Programmierumgebung MATLAB entwickelte Software MATeX www.matex-online.de so erweitert, dass sie Universal 3D – Dateien bei bestimmten Vorgaben erzeugen kann. Mit MATeX lassen sich nun im Handumdrehen nicht nur mathematische Aufgaben lösen, sondern diese zusätzlich mit Universal 3D – Abbildungen anreichern, siehe Abbildung.

Erstellung von PDFs

PDF wurde automatisch generiert

Symbolische Mathematik

Aufgabe wurde analytisch mit MATeX gelöst



Aufgabe zur Höheren Mathematik
Generated by MATEX & Aschenputtel
17. Dezember 2018

Aufgabe 1 (Extrema in 2D mit Nebenbedingungen). Bestimmen Sie alle Extrema der Funktion $f : \mathbb{R}^2 \to \mathbb{R}$ gegeben durch

$$f(x,y) = x^2 - 2x + y^2 - 4y + 5$$

unter der Nebenbedingung

$$h(x,y) := (x-2)^2 + y^2 - \frac{5}{4} = 0$$

mit der Lagrange'schen Multiplikatorenregel.

Ergebnis: Max. bei $(\frac{5}{2}, -1)$; Min. bei $(\frac{3}{2}, 1)$;

Lösung: Die Lagrange-Funktion lautet hier

$$\begin{aligned} F(x,y,\lambda) &= f(x,y) + \lambda h(x,y) \\ &= \lambda\left((x-2)^2 + y^2 - \frac{5}{4}\right) - 4y - 2x + x^2 + y^2 + 5, \end{aligned}$$

mit dem Lagrange-Multiplikator λ. Zu den kritischen Punkten für globale Extrema von f gehören die stationären Punkte der Lagrange-Funktion $F(x,y,\lambda)$, d.h. Punkte, die der Gleichung

$$\nabla F(x,y,\lambda) = \vec{0}$$

genügen. Um diese zu finden, hat man das folgende homogene Gleichungssystem zu lösen:

$$\begin{cases} \partial_x F(x,y,\lambda) = 2x + \lambda(2x-4) - 2 = 0 \\ \partial_y F(x,y,\lambda) = 2y + 2\lambda y - 4 = 0 \\ \partial_\lambda F(x,y,\lambda) = (x-2)^2 + y^2 - \frac{5}{4} = 0 \end{cases}.$$

Dieses Gleichungssystem hat 2 Lösungen:
$(x_1, y_1, \lambda_1) = (\frac{3}{2}, 1, 1)$; $(x_2, y_2, \lambda_2) = (\frac{5}{2}, -1, -3)$;
Weitere kritische Punkte sind die Stellen, an denen die Jacobimatrix der Nebenbedingung

$$\frac{\partial h(x,y)}{\partial(x,y)} = \left(\frac{\partial h}{\partial x}, \frac{\partial h}{\partial y}\right) = (2x-4,\ 2y)$$

keinen vollen Rang hat und die Nebenbedingung selbst erfüllt ist. Dies entspricht hier den Bedingungen

$$\begin{cases} (2x-4,\ 2y) &= (0,\ 0) \\ (x-2)^2 + y^2 - \frac{5}{4} &= 0 \end{cases}.$$

Dieses Gleichungssystem hat keine Lösung und damit gibt es keine weiteren kritischen Punkte.

1

Wir berechnen die Funktionswerte von f an allen kritischen Punkten:
$f(x_1, y_1) = \frac{5}{4}$; $f(x_2, y_2) = \frac{45}{4}$;
Weil durch die Nebenbedingung eine kompakte Menge definiert wird, nimmt die stetige Funktion f auf dieser Menge ihr Minimum und ihr Maximum an.
Das Maximum wird demnach angenommen in:
$(x_2, y_2) = (\frac{5}{2}, -1)$.
Das Minimum wird demnach angenommen in:
$(x_1, y_1) = (\frac{3}{2}, 1)$.

2

Interaktive Abbildung

In Adobe Acrobat Reader interaktiv erkundbar

Buttons für Animation

Rotation, Einblenden von Zusatzinfo

Ziele

- Medienkonvergenz: Anreicherung von Lehrunterlagen mit interaktiven Funktionen
- Automatisches Erstellen von U3D-Abbildungen für ausgewählte Themenbereiche der Mathematik
- Automatisches Einbetten der Abbildungen in PDFs
- Attraktivere Lehrveranstaltungen
- Zugang zum Tool auch für Studierende
- Zum Bedienen keine MATLAB-Kenntnisse notwendig

Umsetzung

- MATLAB-Quelltexte
- Symbolic Math Toolbox zum Umsetzen von analytischen mathematischen Operationen
- Software MATeX zum Lösen von Aufgaben und Erstellen von LaTeX-Dokumentation
- Konverter für U3D-Abbildungen
- JAVA-SCRIPT für Buttons im PDF
- Adobe Acrobat Reader mit aktivierter 3D-Funktionalität

Die Autoren danken der Leitung der Fakultät Grundlagen bzw. dem Mathematischen Institut für die Förderung des Projektes.

Weitere Infos unter: https://www2.hs-esslingen.de/~aeich/digitalerrueckenwind/interaktiv3d/

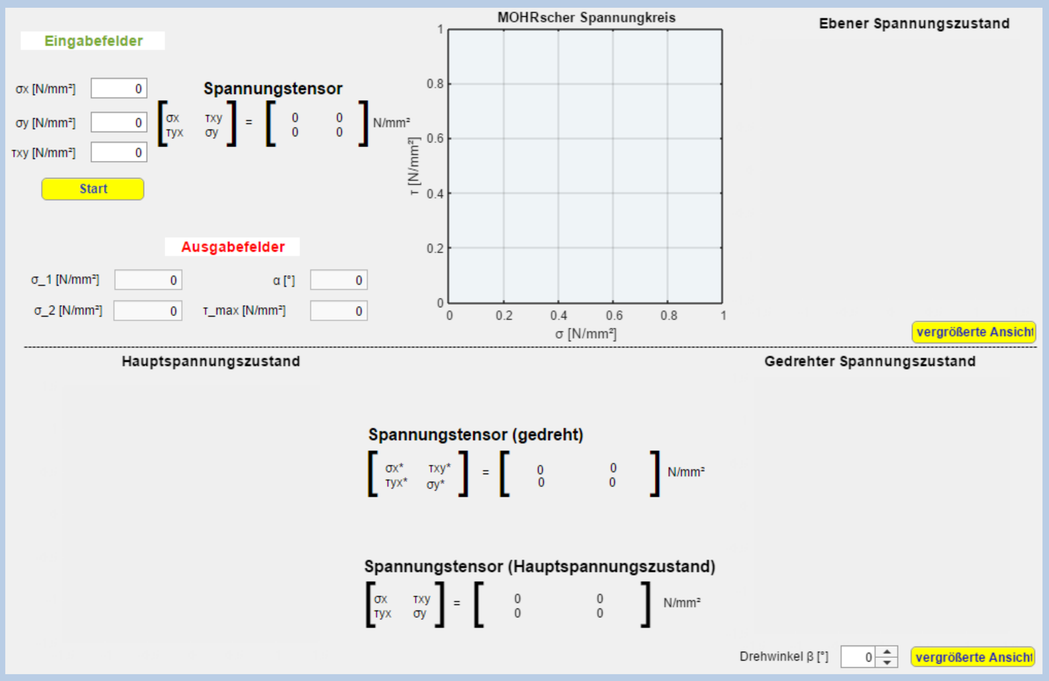
Vorlesungsunterstützung durch MATLAB im Bereich Technische Mechanik

Simon Alber, Karla Ruiz, Carmen Furch

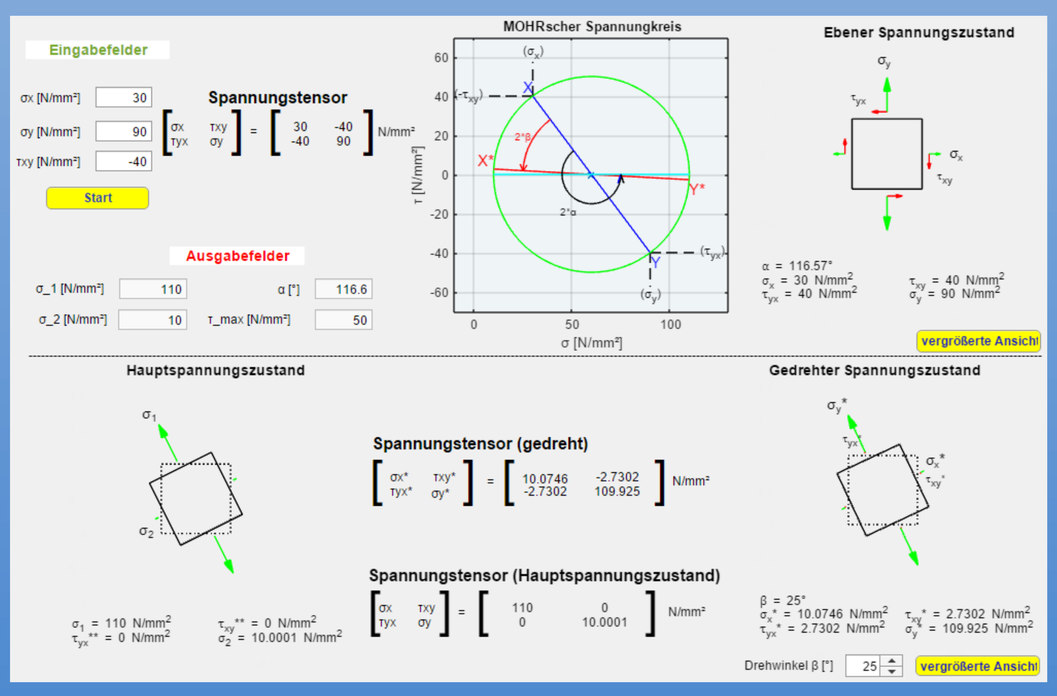
Hochschule für Technik, Wirtschaft und Gestaltung Konstanz

Visualisierung des Mohr'schen Spannungskreises

Liegen in einem dreidimensionalem Bauteil Spannungen in zwei von drei Raumrichtungen vor, so spricht man von einem ebenen Spannungszustand. Hier treten meistens Normal- (σ) und Schubspannungen (τ) auf.
Mit Hilfe des Mohr'schen Spannungskreises können die Hauptspannungen σ_1 und σ_2 zeichnerisch ermittelt werden. Die vorgestellte MATLAB-App soll diese Ermittlung veranschaulichen.
Ist die Belastung am Körper gegeben, so sind auch die auftretenden Normal- und Schubspannungen bekannt. Zunächst müssen diese in die Eingabefelder der App (oben links) eingetragen werden. Soll der Spannungszustand am gedrehten Flächenelement betrachtet werden, so kann im Eingabefeld „Drehwinkel β" (unten rechts) der Verdrehwinkel eingestellt werden.



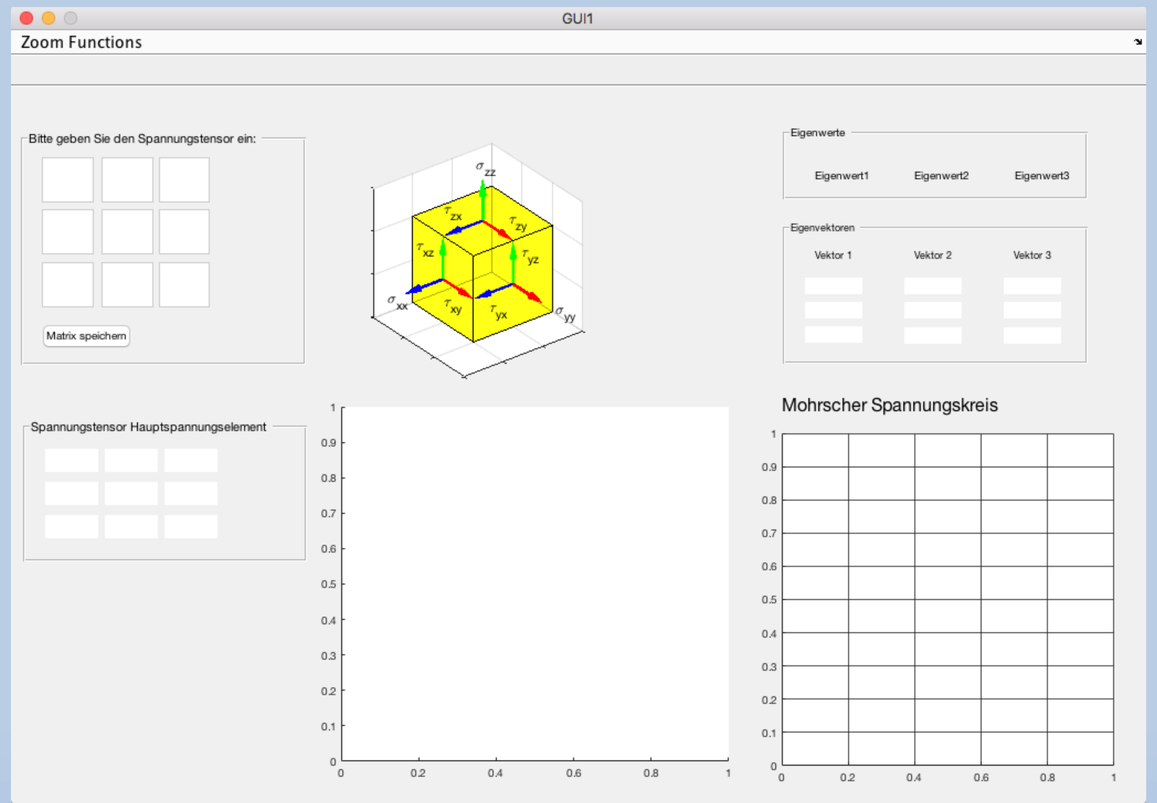
Nach der Eingabe der Spannungen wird durch Klicken des „Start"-Buttons die Visualisierung gestartet. Die drei Flächenelemente oben rechts, unten links und unten rechts stellen den Lageplan in den verschiedenen Zuständen dar. Diese Lagepläne können auch vergrößert betrachtet werden. Der Lageplan des ebenen Spannungszustands oben rechts stellt verhältnismäßig mit der Größe des Pfeiles die eingegebenen Spannungen dar. In Grün werden die Normalspannungen gezeigt und in Rot die Schubspannungen.
Oben in der Mitte wird der Mohr'sche Spannungskreis dargestellt. In Blau sind die Punkte des ebenen Spannungszustands zu sehen, und in Rot die im gedrehten Fall. Für den gedrehten Fall werden von der App die beim gewählten Drehwinkel herrschenden Normal- und Schubspannungen ausgegeben. Durch Verschieben der *X*-, *Y*-Punkte am Mohr'schen Spannungskreis mit dem doppelten Drehwinkel β können diese auch abgelesen werden.
Zum einen stehen in den Ausgabefeldern die ermittelten Hauptspannungen σ_1 und σ_2. Sie charakterisieren den Hauptspannungszustand, bei dem die Schubspannungen gleich null sind. Daher können die Werte für σ_1 und σ_2 an den Schnittpunkten des Mohr'schen Spannungskreises mit der τ-Achse abgelesen werden.
Zum anderen stehen in den Ausgabefeldern der ermittelte Winkel α, welcher angibt, um wieviel Grad das Flächenelement gedreht werden muss, damit dieses sich im Hauptspannungszustand befindet und der Wert von τ_{max}. Er bezeichnet die maximal auftretende Schubspannung. Dieser Wert entspricht dem höchsten Punkt des Mohr'schen Spannungskreises. Die ermittelten Werte kennzeichnen den Spannungstensor, den die App in Matrizenform darstellt.



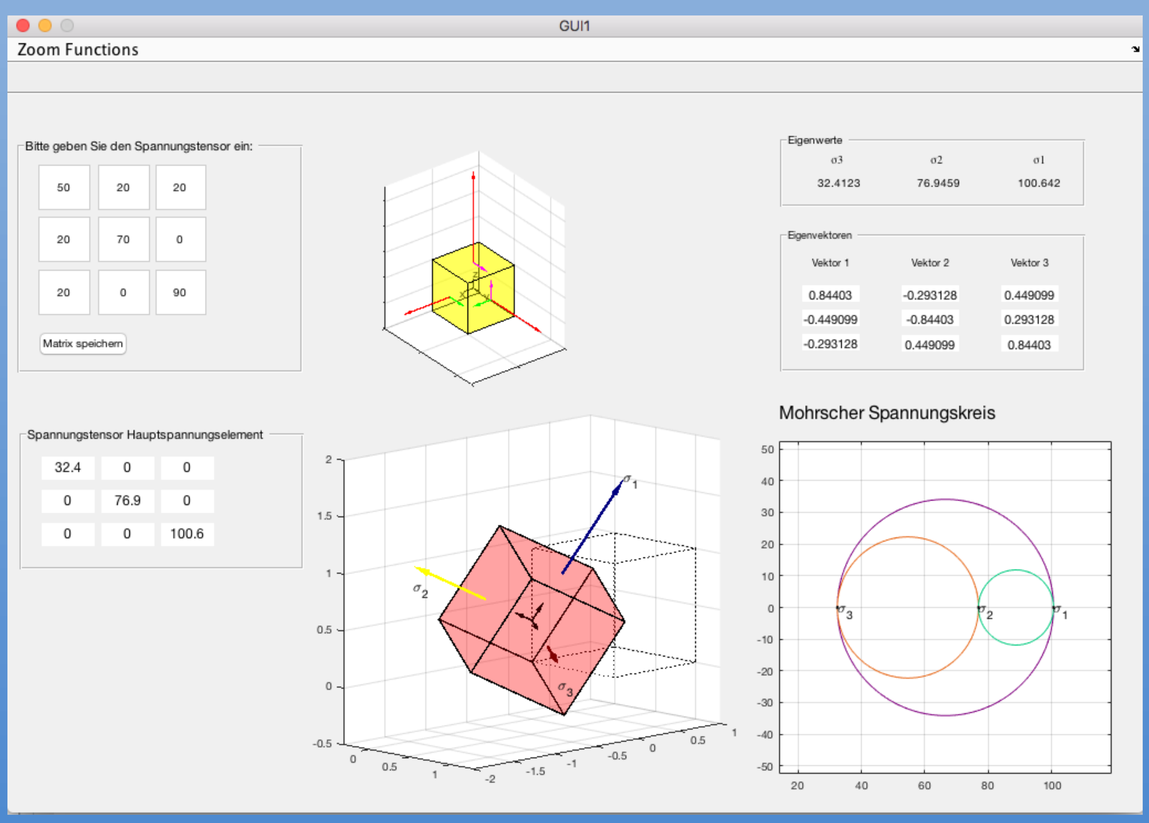
Die App soll als Unterstützung der Vorlesung Technische Mechanik 2 (Festigkeitslehre) an der Hochschule Konstanz dienen. Hierbei soll das Verständnis der Studenten verbessert sowie eine schnelle Visualisierung des Vorlesungsstoffes ermöglicht werden.

Visualisierung des dreidimensionalen Spannungszustandes

Der dreidimensionale Spannungszustand ist durch Spannungen in allen drei Richtungen des Koordinatensystems in einem dreidimensionalen Bauteil gekennzeichnet. Im folgenden MATLAB-Programm soll der Hauptspannungszustand eines Volumenelementes visuell dargestellt werden. Im Hauptspannungszustand eines Punktes im Bauteil treten nur Normalspannungen σ und keine Schubspannungen τ auf. Die σ-Werte hängen von der gegebenen Belastung ab.
Dazu muss zunächst der aus der bestehenden Belastung resultierende Spannungstensor eingegeben werden. Der gelbe Würfel dient als Referenz. Die farbigen Spannungen zeigen, in welche Richtung die jeweils eingegebenen Normalspannungen σ_{xx}, σ_{yy} und σ_{zz} sowie die Schubspannungen wirken.



Durch Klick auf den Button „Matrix speichern" werden die Spannungen im gelben Referenzwürfel entsprechend des eingegebenen Betrags angepasst. Die Eigenwerte und ihre zugehörigen Eigenvektoren werden berechnet und rechts ausgegeben. Der rote Würfel zeigt den in den Hauptspannungszustand gedrehten Referenzwürfel. Die dazugehörigen Eigenspannungsvektoren sind in Rot, Blau und Gelb dargestellt. Im Mohr'schen Spannungskreis rechts daneben ergeben sich die Farben orange, grün und violett jeweils aus der Mischung der den Eigenspannungen zugeordneten Farben.
Unten links wird der aus den berechneten Eigenwerten bestehende Spannungstensor des Hauptspannungszustandes ausgegeben. Die dreidimensionalen Darstellungen können manuell gedreht werden, um ggf. eine bessere Perspektive zu erlangen. Außerdem kann jede Darstellung über den Reiter ZOOM FUNCTIONS vergrößert und verkleinert sowie bewegt werden.
Das Programm prüft, welche Spannungen negativ und welche positiv sind, da negative Spannungselement nicht in den Würfel hineinzeigen sollen, sondern mit der Pfeilspitze von außen auf die Würfelflächen deuten sollen.



Das Programm soll zur Vorlesungsunterstützung im Bereich Technische Mechanik an der Hochschule Konstanz dienen und durch die Visualisierung zum besseren Verständnis des zwei- und dreidimensionalen Hauptspannungszustandes beitragen.

MATLAB EXPO 2019

Hochschule Esslingen
University of Applied Sciences

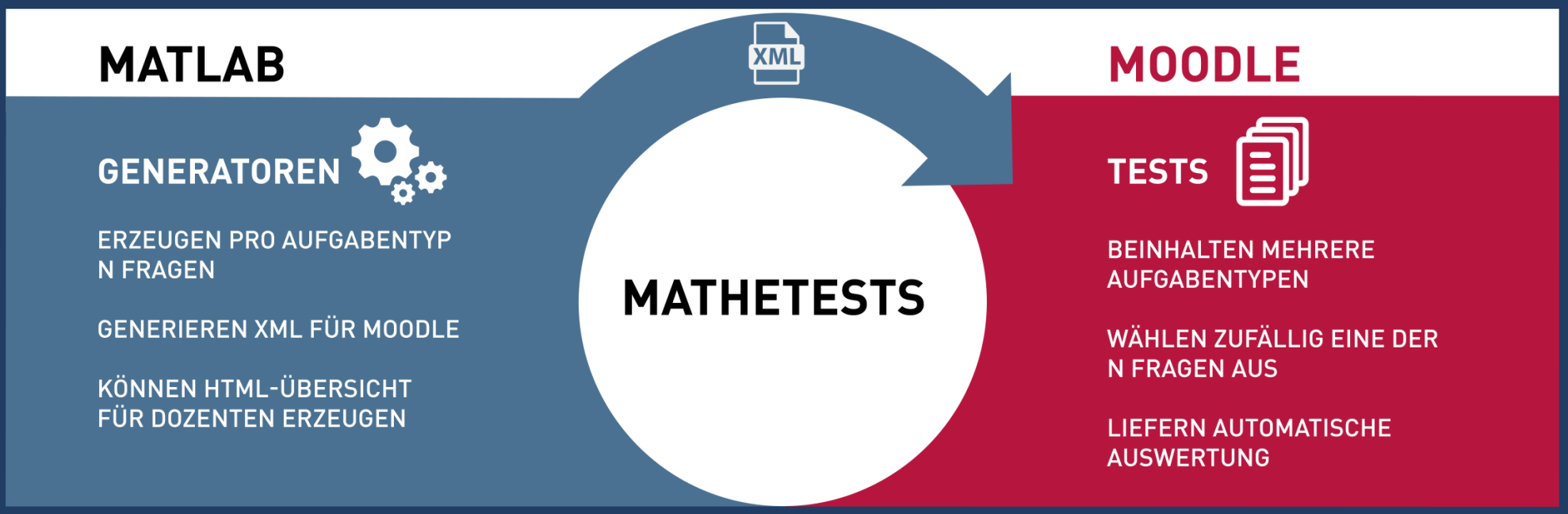
DIGITALER RÜCKENWIND FÜR MATHEMATIKVERANSTALTUNGEN

Automatisches Generieren, Durchführen und Auswerten von Mathematiktests in Moodle

Prof. Dr. Gunther Schaaf, Dipl. Inf. Achim Eichhorn, Prof. Dr. Andreas Helfrich-Schkarbanenko, Fakultät Grundlagen

Im WiSe 2018/19 haben wir unser Lernarrangement der Vorlesung Mathematik I um Tests im Sinne von **Blended Learning** erweitert. Hierfür wurden **MATLAB-Skripte** entwickelt, die **Testaufgaben** für ausgewählte Fachthemen zufällig generieren, diese lösen und für das unmittelbare Einbetten in **Moodle** in XML-Format vorbereiten.

Dabei werden die Möglichkeiten der Lernplattform voll ausgereizt. Das selbstgestaltete Üben, die Durchführung sowie die automatische Auswertung der Tests finden in Moodle statt. So lassen sich hochflexibel und zeitsparend abwechslungsreiche Übungen und Tests auch zu Themen der **Physik** und **Elektrotechnik** aufstellen.



Ziel

- Wissen über mathematische Grundlagen festigen
- Wiederholen der Lerninhalte **während** des Semesters
- Erworbene Kenntnisse durch Tests nachweisen
- Dabei darf (und soll) beliebig oft geübt werden

Vorteile

- Individueller zufälliger Test bei jedem Versuch
- Unmittelbares Feedback erlaubt direkte Lernerfolgskontrolle
- Kein Korrekturaufwand: Entlastung von Dozent und Tutor
- Unkompliziertes Aufgabendesign – nur MATLAB-Kenntnisse nötig



Steckbrief – Stand März 2019

- 5 Tests zu 9 verschiedenen Themen wie z.B. Bruchrechnung, Quadratische Gleichungen, Vektorrechnung, Potenzgesetze, Funktionen, Differenzialrechnung
- 51 Aufgabentypen
- Über 1400 Fragen insgesamt generiert
- Schnelle Qualitätskontrolle im Übersichtsdokument (HTML)

Ausblick

- Produktiver Einsatz im Rahmen von Mathematikveranstaltungen verschiedener Studiengänge
- Beliebig erweiterbar hinsichtlich Aufgabenanzahl und Themenspektrum
- Erweiterbar auf andere Ausgabeformate (LaTeX/PDF, andere LMS, …)

Weitere Infos unter: www2.hs-esslingen.de/~aeich/digitalerrueckenwind/aufgabengeneratoren/

WWW.HS-ESSLINGEN.DE/
Hochschule Esslingen // Kanalstraße 33 // 73728 Esslingen // Tel +49(0)711.397-000 // info@hs-esslingen.de

Zertifikat seit 2006 audit familiengerechte hochschule
EMAS GEPRÜFTES UMWELTMANAGEMENT DE-175-00173
Design: www.weiser-design.de // Satz: Hochschule Esslingen

Das Elektronik Praktikum in der eigenen Bude, Digitaltechnik mit Simulink als Telematik Versuch.

Jochen Theis, Günter Schmitz FH Aachen University of Applied Sciences

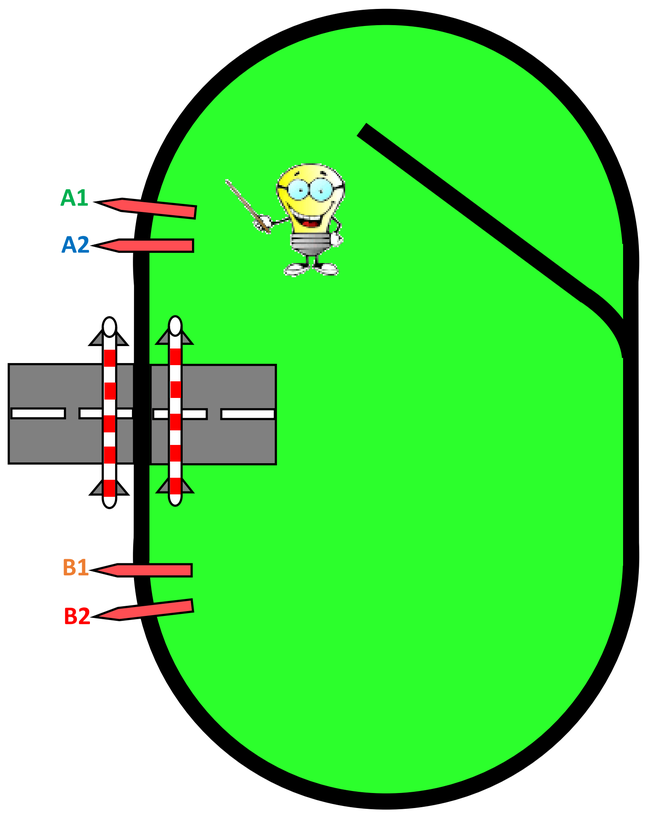
FH AACHEN UNIVERSITY OF APPLIED SCIENCES

Einleitung und Motivation

In der Lehrveranstaltung **Elektronik**, Pflichtfach Bachelor im Fachbereich **Luft– und Raumfahrttechnik**, werden die Grundlagen der Halbleiter und Digitaltechnik vermittelt. Das **Praktikum** der **Elektronik** findet als **Telematik Versuch** statt. Die Studierenden führen die Versuche zum Thema **Digitaltechnik, Transistor und Operationsverstärker** nicht mehr im Labor als Anwesenheitspraktika durch, sondern **von einem PC, der sich im Internet befindet**. In einem Versuch werden **reale Messdaten (z.B. die Verstärkerkennlinie eines Transistors) gemessen und ausgewertet**. Abschließend findet eine **Besprechung** des Versuches **im Labor** statt. Für den **Digitaltechnik Versuch** wird u. a. eine **Steuerungslogik einer Modelleisenbahn** entwickelt und getestet. Dazu ist die **Schnittstelle der Eisenbahnsteuerung** erweitert worden, damit die **Programmierung der digitalen Logik** mit **Simulink** und **Stateflow** umgesetzt werden kann.

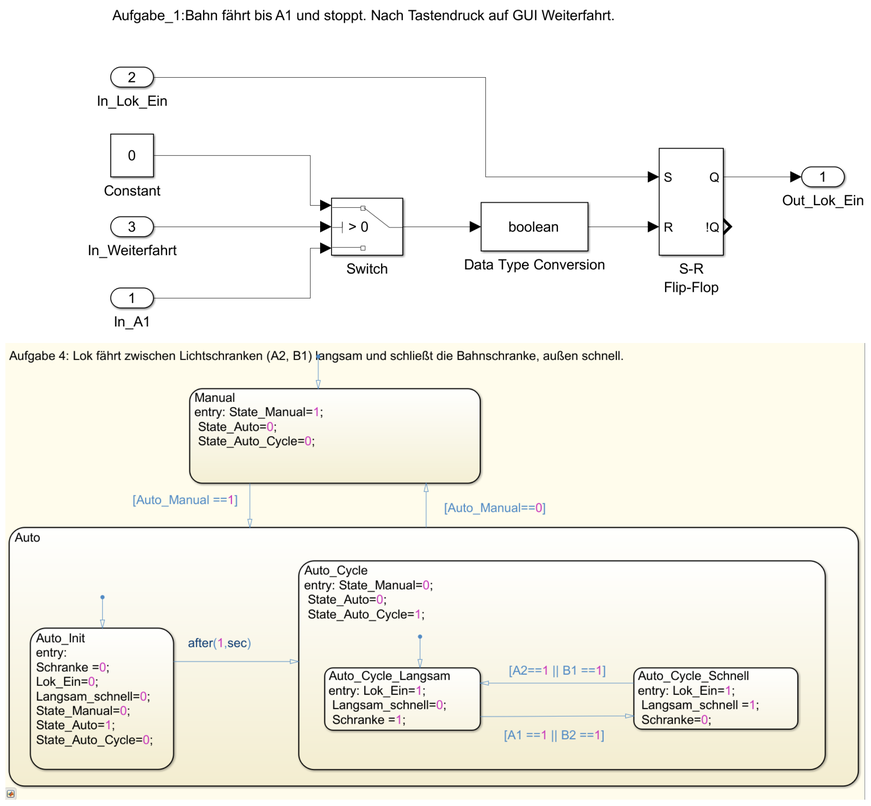
Übersicht des Telematik- Versuches der Eisenbahn

- Es gibt **vier Signale aus dem Modell**, um den Eisenbahnversuch zu steuern: **Lok_Ein, Geschwindigkeit (langsam– schnell), Richtung, Bahnschranke**
- Es gibt **vier Signale, die von der Eisenbahn Controller zurück gemeldet werden** und im Modell ausgewertet werden: **Lichtschranken A1, A2, B1, B2**
- Eine **Webcam** überträgt **live** die Position der Lokomotive und Schranke
- Bei **Dunkelheit** schaltet sich eine **Beleuchtung** ein
- Die **Steuerungssoftware der Eisenbahn überwacht** den **Zugriff**, **empfängt** die **Steuerungsbefehle** und **versendet** die **Rückmeldungen** (z.B. Status der Lichtschranken, Zugriffszeit, etc.) an den **Client**.



Technische Umsetzung

- In **Simulink** ist ein **Framework** erstellt worden, wo die **Eingänge** und **Ausgänge** der Eisenbahn **vorhanden sind**.
- Der **Studierende ergänzt das Modell**, um die vorgegebene Aufgabe zur Digitaltechnik zu lösen.
- Das **erstellte Modell** wird mittels **C- Codegenerierung für ein Echtzeittarget** in einer (Realtime)- Controller App implementiert.
- Diese **Controller– App** baut die Verbindung zum **Eisenbahn Controller über TCP-IP** auf.
- Die **Funktionen des Modells** können **live betrachtet, getestet und optimiert** werden.





Zusammenfassung

Durch die **Erweiterung der Schnittstelle des Telematik Versuches „Eisenbahn**“ können **zukünftig Aufgabenstellungen der Digitaltechnik** in **Simulink und Stateflow** umgesetzt werden. Für die Studierenden bietet sich bei diesem **Telematik– Praktikum der Vorteil**, den Versuch **überall und jederzeit zu absolvieren** (Internetanschluss vorausgesetzt!). Aus Sicht des Labors wird nur ein Versuchsplatz zur Verfügung gestellt und gewartet, an dem **ca. 200 Studierende** im Sommersemester ihr **Praktikum absolvieren** können.

Die **Telematik Versuche in der Elektronik** sind **sehr beliebt** bei den Studierenden und leisten einen **wertvollen Beitrag** zum Thema **eLearning**.

Rupprecht-Gymnasium München
Naturwissenschaftlichtechnologisches
und Sprachliches Gymnasium

ZUSAMMEN.WACHSEN.

Individualisierte Begabtenförderung mit MATLAB® & Simulink® – Gezielte MINT-Förderung an der Schnittstelle Schule-Universität

Beobachtung

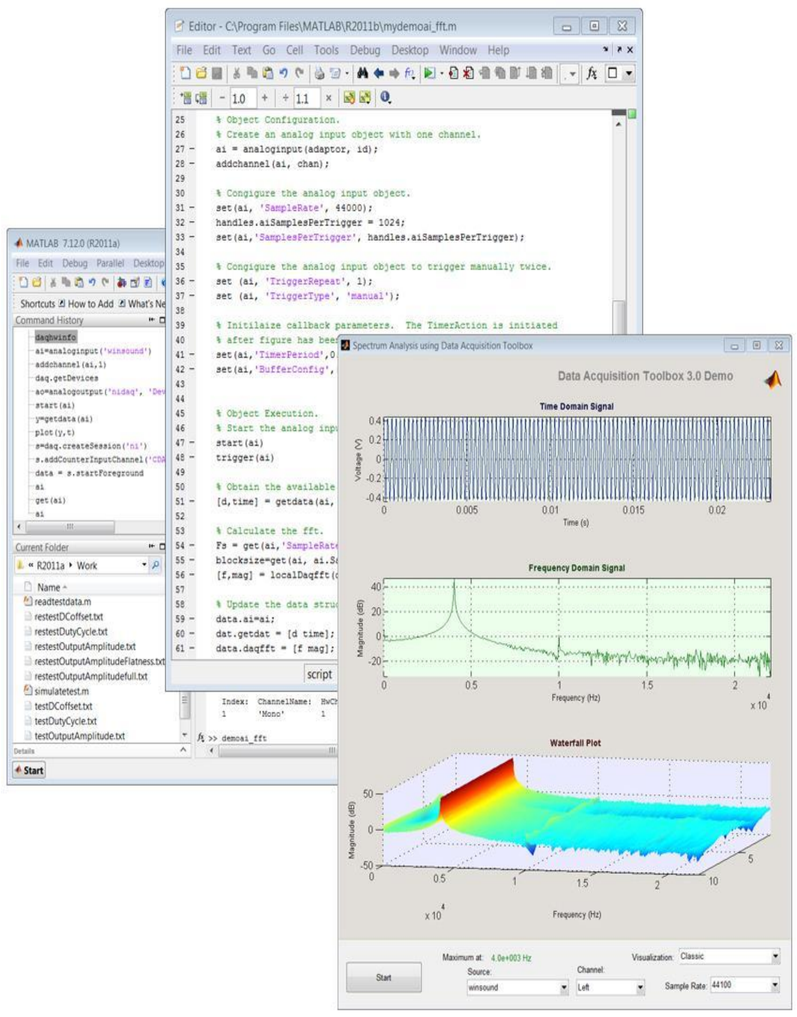
- Zunehmende Bedeutung von mathematischer Modellierung und Simulation mit wachsender Rechenleistung und Automatisierung von Prozessen in den unterschiedlichsten Anwendungsbereichen

Zielvorstellung

- Frühzeitiger Zugang zu mathematischer Modellierung und Simulation
- Gezielte, individuelle Förderung begabter Schülerinnen und Schüler im MINT-Bereich entsprechend ihres überdurchschnittlichen Potentials
- Zukunftsgerichtete Berufsorientierung
- Sprungbrett zu Forschung und Entwicklung in Universität und Industrie

Ansatz

- Selbstgesteuerter, langfristig angelegter, kumulativer Lernprozess
- Gewährleistung kontinuierlich hoher intrinsischer Motivation
- Sukzessives Wachstum an selbst gesetzten Herausforderungen
- Leistungsstarke Werkzeuge, die möglichst keine Grenzen nach oben setzen



Mehrstufige Umsetzung

- Research Days
- Wissenschaftspropädeutisches (W-)Seminar der bayerischen Oberstufe
- Austausch mit der internationalen MATLAB® & Simulink® Support- und User-Community
- MINT-EC Schülercamp
- Wettbewerb Jugend forscht

Research Days

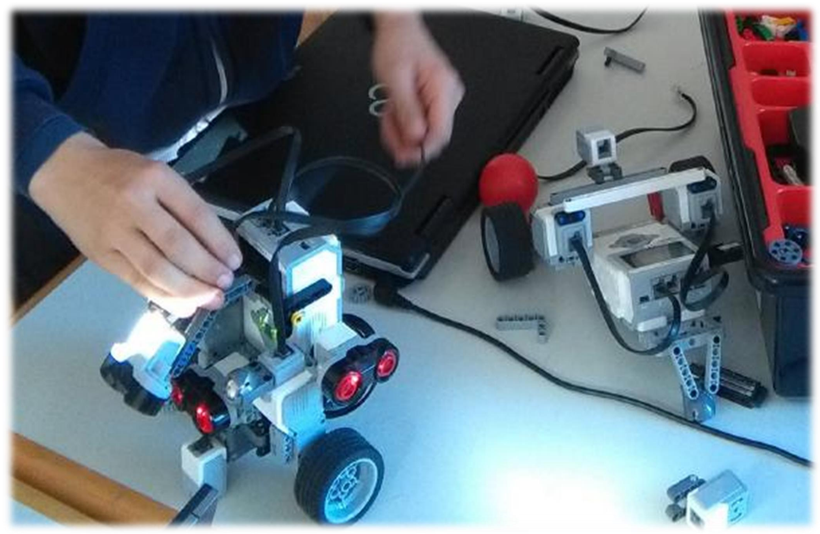
- Einführungstage zum Kennenlernen und Umgang mit MATLAB & Simulink
- Gemeinsame Hackathon-Veranstaltungen für Fortgeschrittene
- Unterstützung durch MathWorks

W-Seminar Modeling and Simulations with MATLAB ® & Simulink ®

- Ziel: Anwendungen mathematischer Modellierung und Simulation
- Softwareeinsatz: MATLAB® und Simulink®
- Bilingual: Recherchen, Seminararbeit und Präsentationen in der Wissenschaftssprache Englisch
- Bisherige Durchführungen:
  - 2016/18: Leitfach Mathematik
  - 2017/19: Leitfach Physik
  - 2018/20: Leitfach Mathematik
- Themenbeispiele:
  - Obstacle Detection and Circumvention Algorithms for a Robot
  - Machine Learning
  - Black Hole Simulation
  - Maxwell Equations
  - Automatic Solving of Sudokus
  - Influenza & Virus Kinematics
  - Portfolio Theory, Portfolio Optimization and Asset Allocation

Austausch mit der internationalen MATLAB & Simulink Support- und User-Community

- Zugriff auf die globale Projekt-Datenbank File Exchange auf MathWorks.com
- Hilfe zur Selbsthilfe: zahlreiche Online-Tutorials und zeitnaher Support
- Austausch in der Wissenschaftssprache Englisch



MINT-EC Schülercamp Robot Challenge

- Selbstgesetze Zielen
- 3 Tage intensive Teamarbeit
- Steuerung von LEGO EV3 ® Robotern mithilfe von Simulink ®
- Ausgewählte, begabte Jugendliche aus verschiedenen MINT-EC-Schulen
- Lernort ohne Ablenkungen: Schullandheim des Rupprecht-Gymnasiums in Holzhausen
- Gefördert durch den vbw
- 4-Phasen Konzept:
  1. Einführungs-Workshop in Simulink
  2. Teambildung und individuelle Projektformulierungen
  3. Umsetzung der Projekte
  4. Live-Demonstrationen und technischer Präsentationen

Wettbewerb Jugend forscht

- Projekte des W-Seminars oder des MINT-EC Schülercamps als Basis
- Zusätzlicher Anreiz für die Schülerinnen und Schüler
- Unterstützung durch MathWorks
- 2. Preis im Landeswettbewerb (Bayern)

Aufbau von Schlüsselkompetenzen

- Erfahrung mit komplexen Fragestellungen
- Nutzung der in Wissenschaft und Industrie weit verbreiteten Standardwerkzeuge MATLAB® und Simulink®
- Selbstverständlicher Umgang mit der Wissenschaftssprache Englisch als Grundvoraussetzung für einen zukünfigten Erfolg in Forschung oder Industrie.

➢ Frühzeitiger Aufbau dieser Schlüsselkompetenzen als wichtiges Karrieresprungbrett

Rupprecht-Gymnasium München
Albrechtstraße 7
80636 München
mint@rupprecht-gymnasium.de

MathWorks
Adalperostr. 45, 85737 Ismaning
www.mathworks.de

MINT-EC
Poststraße 4/5, 10178 Berlin
info@mint-ec.de

Mobiler Temperaturregelungsversuch

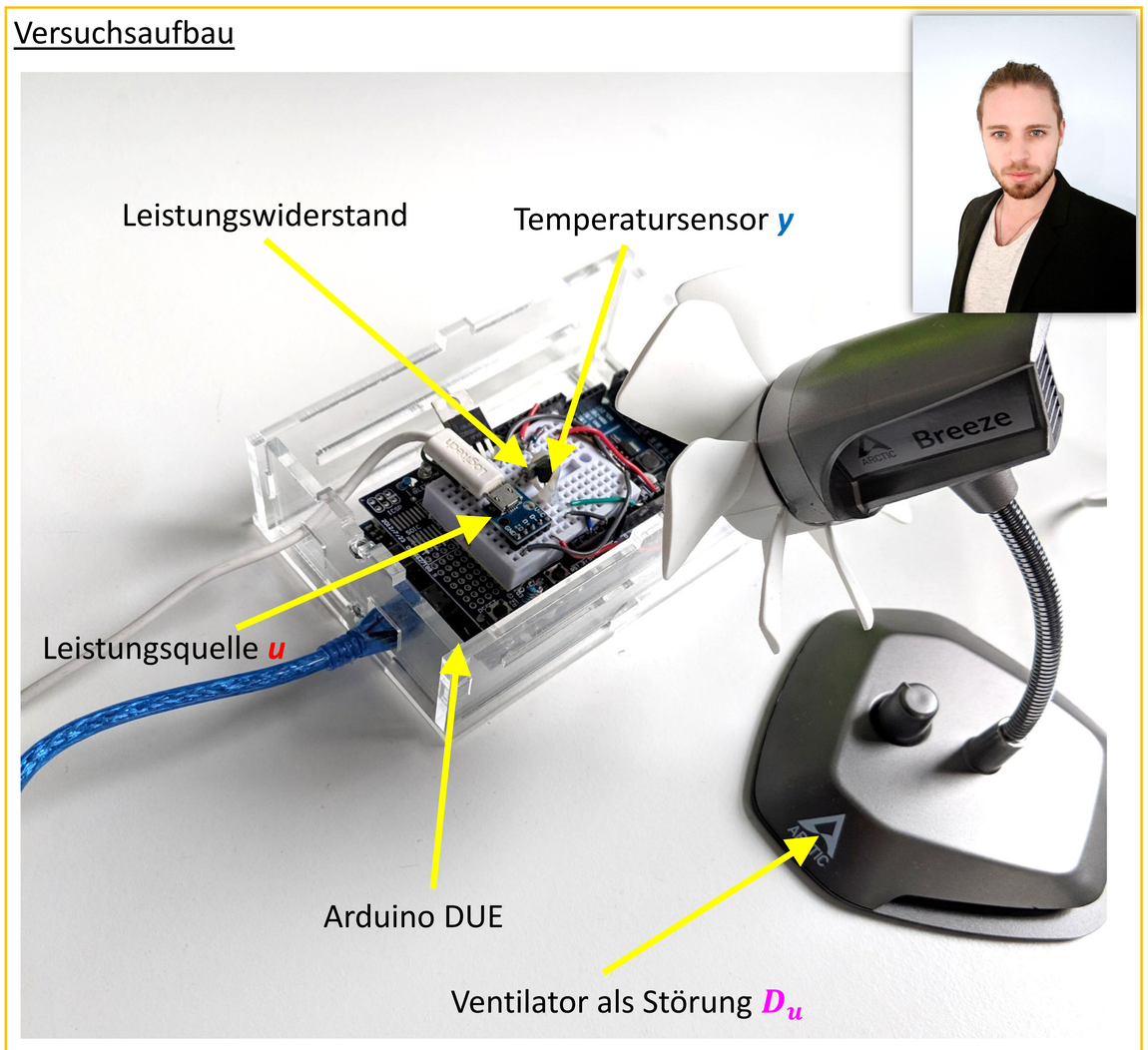
AFE-Master, Projektarbeit I + II

Labor für Regelungstechnik (Prof. Dr. Peter Zentgraf), Raum D 2.10, Sebastian Fischer

Technische Hochschule Rosenheim
Technical University of Applied Sciences

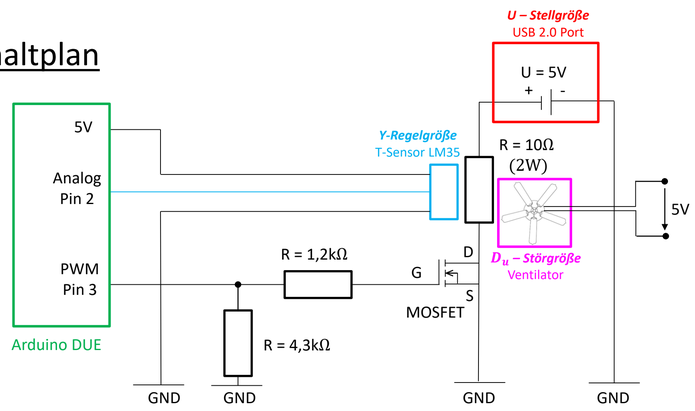
Stand: 25.02.2019

Versuchsaufbau



Der Versuchsaufbau besteht aus einem zu erwärmenden Leistungswiderstand (10 Ohm, 2 Watt), einem analogen Temperaursensor LM35cz, einer USB Leistungsquelle sowie einem Arduino DUE + Protoshield (inkl. Steckbrett). Die auf dem Steckbrett bestehende elektrische Schaltung ist in der Abbildung unten dargestellt.

Elektrischer Schaltplan



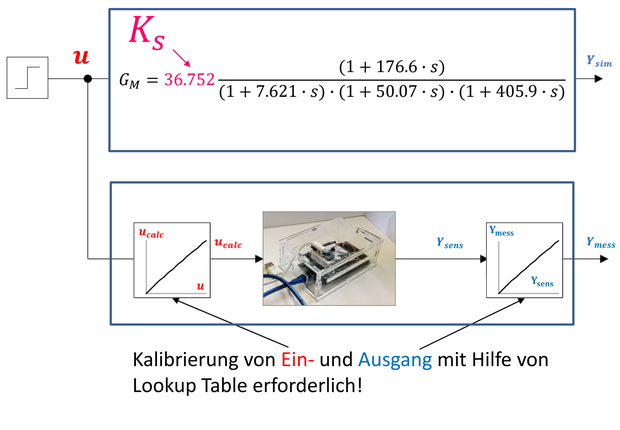
Ziel der Arbeit

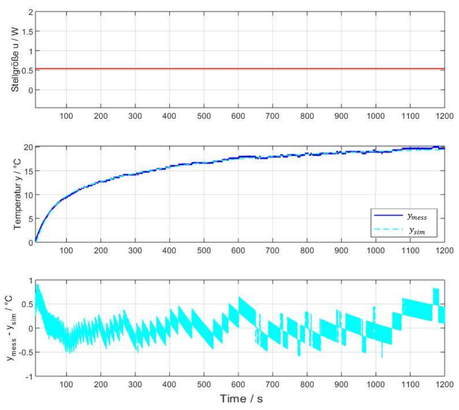
Mit dieser Projektarbeit soll der grundsätzliche Unterschied zwischen Steuerung und Regelung anhand eines zu erwärmenden Leistungswiderstandes erklärt werden. Als Stellgröße dient dabei die elektrische Leistung welche den Widerstand erwärmt. Die Ausgangsgröße ist durch die an einem analogen Temperatursensor gemessene Temperatur beschrieben.

Dazu wird eine Übertragungsfunktion experimentell mit Hilfe von pzMove [1] erstellt, welche als Grundlage für alle folgenden Modell Simulationen dient.

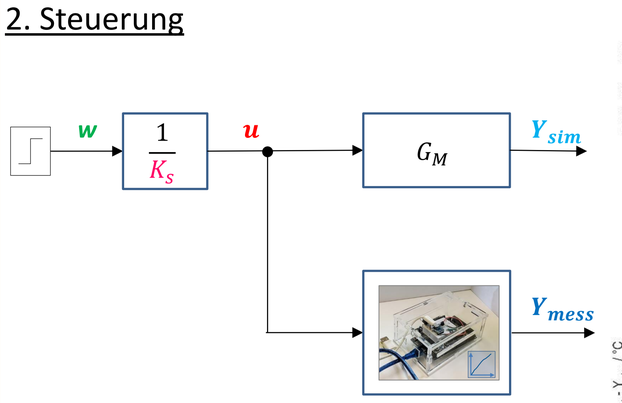
Bei allen Modellen soll der Widerstand um 10K erwärmt werden.

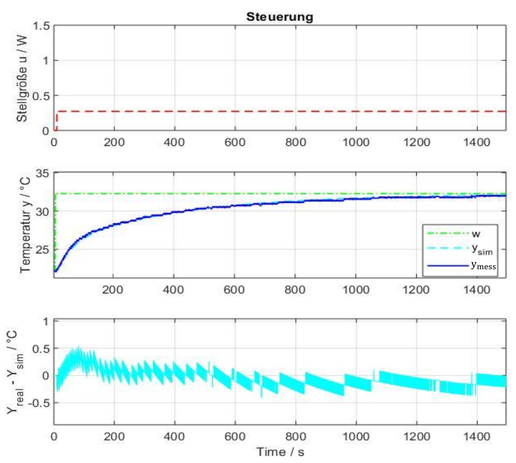
1. Modellierung der Regelstrecke als PT3-System [2]





2. Steuerung

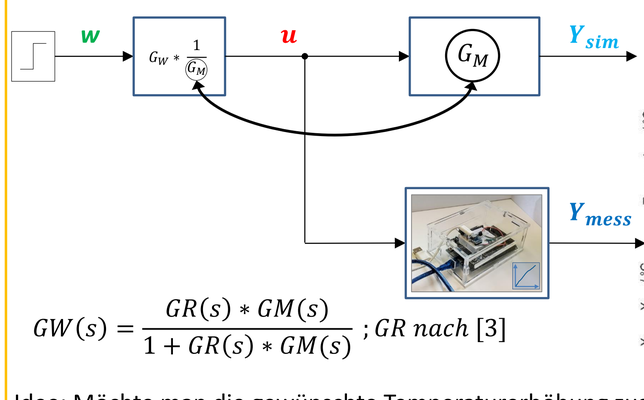




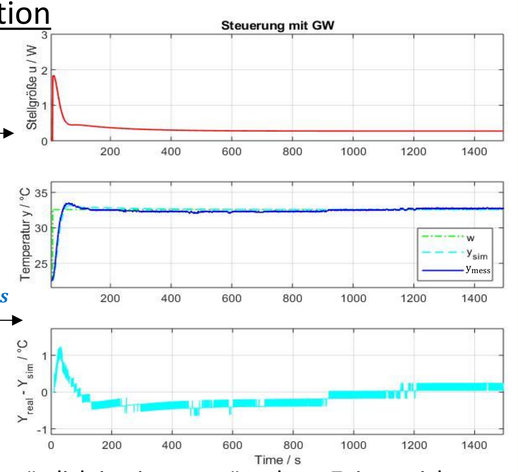
Idee: Eine gewünschte Temperaturerhöhung lässt sich durch einfache Steuerung realisieren. Die erforderliche Zeit hängt von der Strecke ab.

3. Steuerung mit Wunschübertragungsfunktion

a) ohne Störung

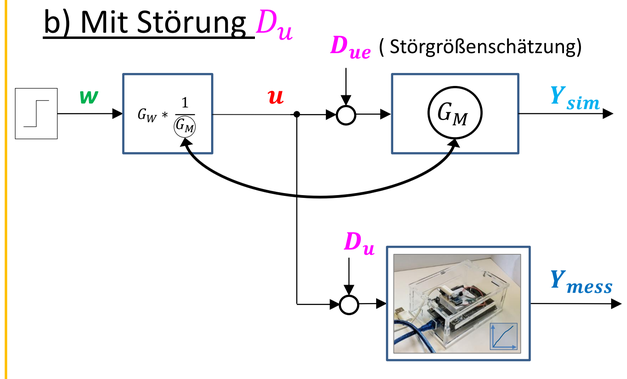


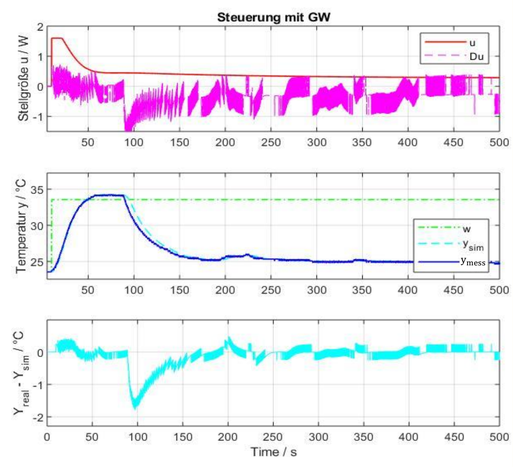
$$GW(s) = \frac{GR(s) * GM(s)}{1 + GR(s) * GM(s)} \; ; GR \; nach \; [3]$$



Idee: Möchte man die gewünschte Temperaturerhöhung zusätzlich in einer gewünschten Zeit erreichen, so lässt sich das ebenfalls ohne Regelung bewerkstelligen.

b) Mit Störung D_u

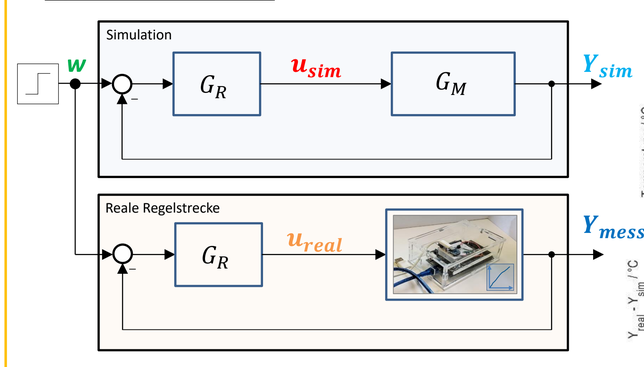


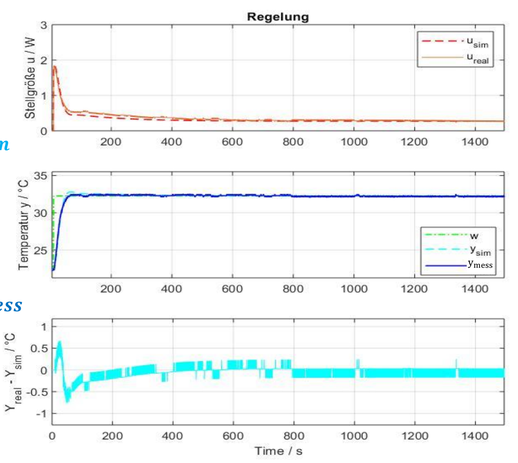


Idee: Gibt man eine Störung auf das System, kann die Steuerung nicht darauf reagieren.

4. Regelung

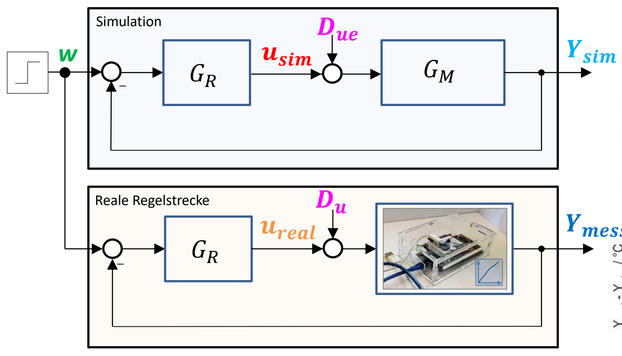
a) Ohne Störung

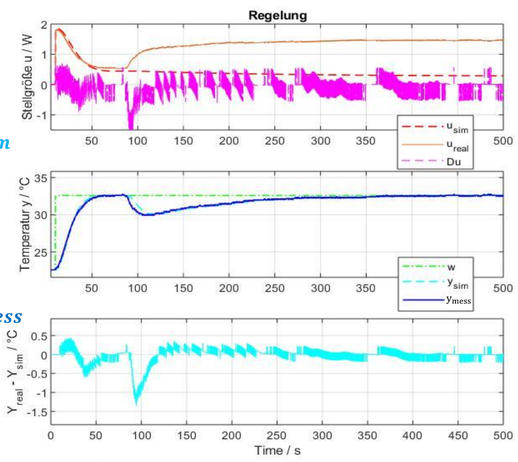




Idee: Vergleicht man die Regelung mit der Steuerung aus 3a, so sind keine Vorteile ersichtlich. Beide Systeme heizen gleich schnell.

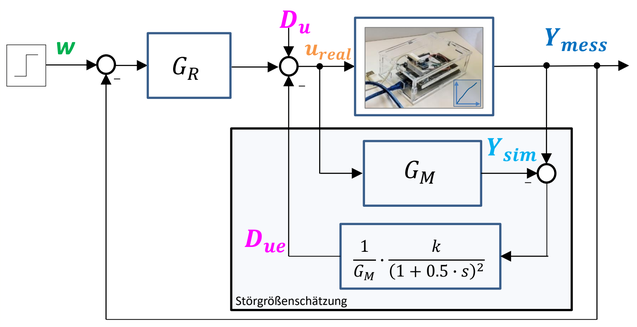
b) Mit Störung D_u

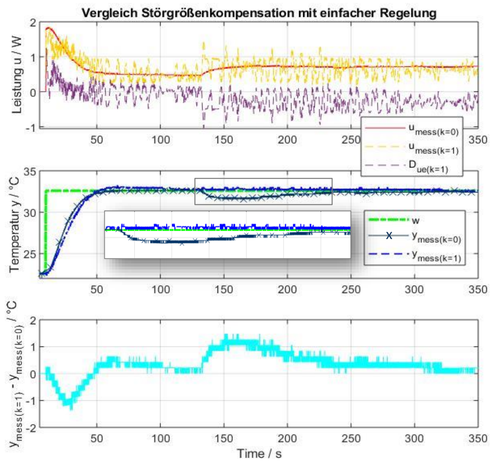




Idee: Möchte man zusätzlich das Aufheizen unempfindlich gegen Störungen machen, wird eine Regelung benötigt.

5. Störgrößenkompensation





Idee: Möchte man den Einfluss der Störung schneller kompensieren, so kann die Störgröße geschätzt und direkt kompensiert werden.

6. Zusammenfassung

Die Steuerung mit Wunschübertragungsfunktion ist gegenüber der einfachen Steuerung wesentlich schneller. Zwischen Steuerung mit Wunschübertragungsfunktion und Regelung ist bzgl. Schnelligkeit kein Unterschied zu erkennen. Erst wenn man eine Störung auf das System gibt, wird der Unterschied zwischen Steuerung und Regelung klar. Während die Regelung auf diese Störung reagiert und die Stellgröße dementsprechend anpasst, reagiert die Steuerung nicht auf die Störung. Dies ist der Fall, da die Regelung eine Rückführung beinhaltet und Soll- an Istwert ständig angleicht. Die Steuerung dagegen ist ein rein vorwärts gerichteter Prozess. Man kann den Einfluss der Störung schneller kompensieren, indem man die Störung schätzt und in eine Stellgröße umwandelt, welche die Störung direkt beim Auftreten kompensiert.

Literaturhinweise:
[1] pz-Move Short-Link, http://www.th-rosenheim.de/rt-pzMove.html
[2] Zentgraf, P., *Generieren einer Streckenübertragungsfunktion aus gemessenen Ein-/Ausgangsdaten*, 16. Konferenz für Angewandte Automatisierungstechnik in Lehre und Entwicklung an Hochschulen (AALE), Heilbronn, 27. Februar - 1. März 2019.
[3] Einstellregeln nach Chien / Hrones / Reswick, Prof. Dr.-Ing. Peter Zentgraf: Regelungstechnik I | Kap. 7; Synthese – SS 2015

Position Control of Brushless DC-Motors in a Low-Cost Laser Deflection Device

Peter Krönes, Ferdinand Friedrich and Christoph Ament
Chair of Control Engineering, University of Augsburg
Ferdinand.Friedrich@informatik.uni-augsburg.de

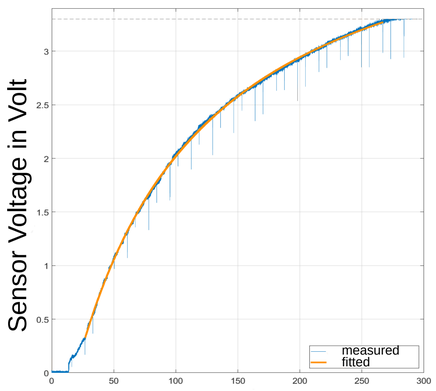
Deflection Systems for laserbeams usually consist of two orthogonally arranged mirrors which rotation is actuated by a galvanometer. Systems that meet the requirements in terms of dynamics and precission are usually expensive. In an attempt to build a laser deflection device at much lower cost, brushless dc-motors are used. The angular position is measured by potentiometer. To eliminate negative effects of the position measurement signal filters are applied. The motor is then identified via the *System Identification Toolbox.* A linear quadratic regulator with integrated output-feedback is used to control the angular position of each motor. Trajectories can be planned in an *App Designer* user interface, which is sending values to the Simulink model. The code for the Nucleo board which is controlling both motors, is generated via automated code-generation. Deflection devices like this can be used for an absolute and non-contact position measurement using interferometers, which is highly relevant for robots like the modular cable robot (MoCaRo).

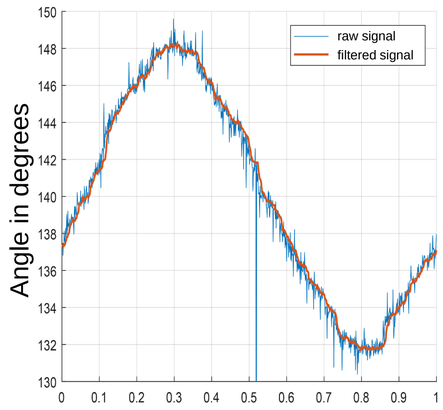
Angle Sensor Characteristics and Filter

For the measurement of an angle with a potentiometer it is necessary to use a potential divider. The relation between voltage and angle can be described as follows:

$$U_m(\varphi) = U_v \cdot \frac{R_{pot}(\varphi)}{R_1 + R_{pot}(\varphi)}$$

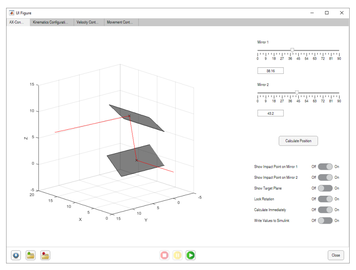
To eliminate manufacturing tolerances, the *Curve Fitting Toolbox*™ is used to get a reliable function.



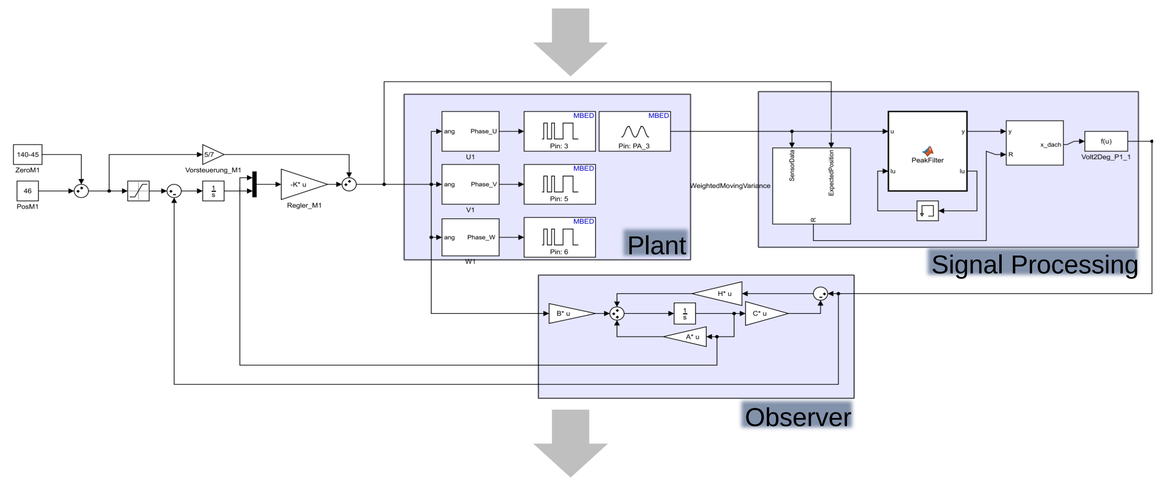


For noise reduction of the potentiometer, a peak filter and a Kalman filter are used. For the Kalman filter the potentiometer is modelled as a linear system without an input.

Control UI, Simulink and Code Generation



The *App Designer* has been used to create a user interface to control either the angles of both mirrors separately or plan a trajectory. The values are transferred to the Simulink model.



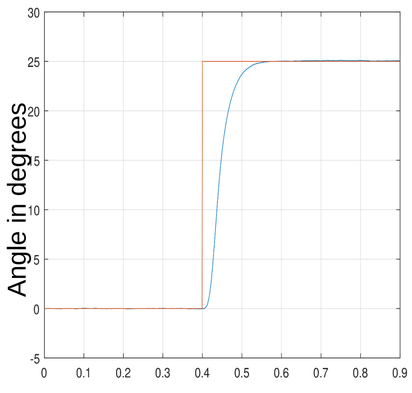
The *Simulink Coder* support package for Nucleo boards is used for automated code-generation, the model is the run in external mode on the Nucleo F767ZI.

Motor Identification and Control Structure

The motor is modelled like a simple dc-motor and it can be assumed that the transfer behavior can be described as follows:

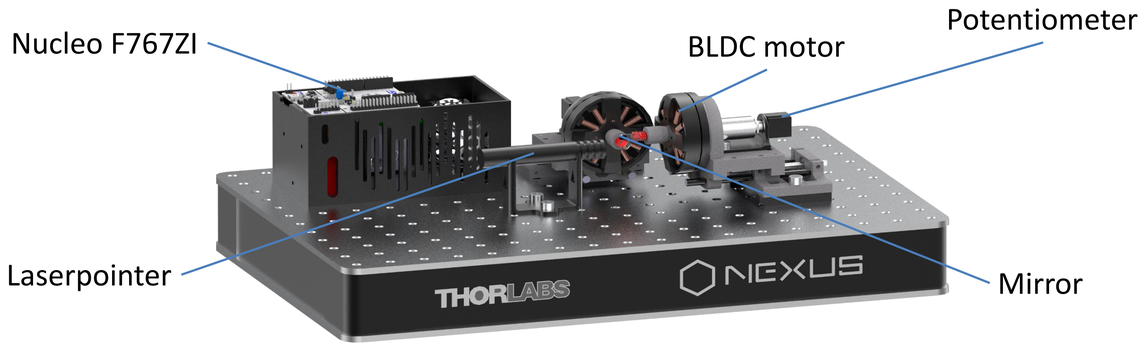
$$G(s) = \frac{b_0}{a_2 s^2 + a_1 s + a_0}$$

As there is not enough information for a theoretical approach, the *System Identification Toolbox*™ is used to analyze the step response of the motor.



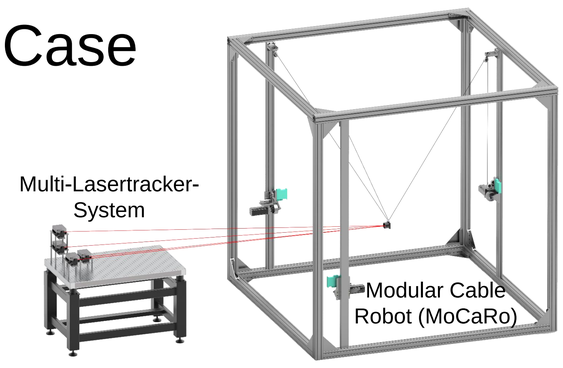
The system is controlled by a state space regulator, which is in need of the system states. As the system itself is closed, an observer is needed. Therefore a Kalman observer is implemented. To control the system without deviation, an integral output-feedback is added. The linear quadratic regulator has been designed for the extended system.

Deflection System

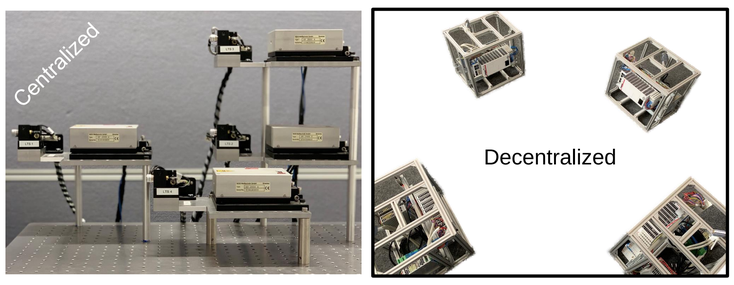


Use Case

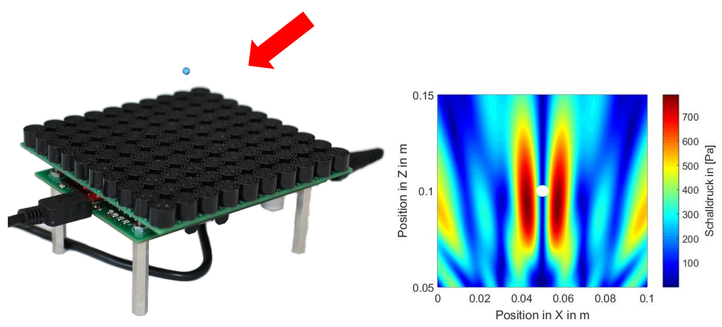
MoCaRo and distributed Lasertracker: optical closed-loop control (RMSE < 20 µm)



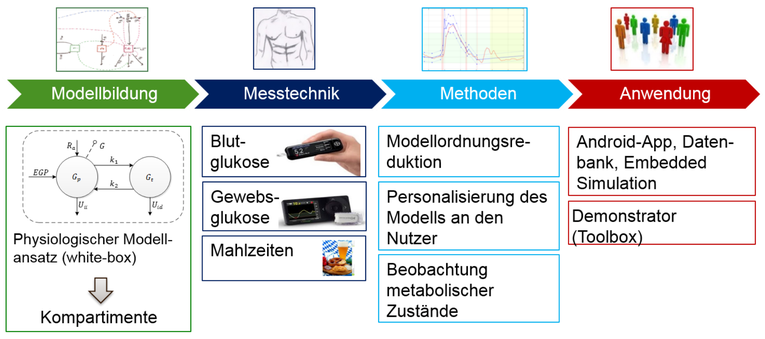
State-of-the-Art Research



Multi-Lasertracker with 4 sensor modules using *Simulink Coder* for automated code-generation: contactless with RMSE < 3 µm @ 1 m/s



Ultrasonic Levitation: Multi actuating & sensing (10x10 piezo array), code-generation for FPGA with specific tools developed for *MATLAB*, real-time control of an instable system.



State- and parameter estimation of dynamic metabolic models for personalized model-based diabetes management; Using *MATLAB* and system engineering methods to support patients with diabetes mellitus.

TECHNISCHE UNIVERSITÄT DRESDEN

Fakultät für Elektrotechnik und Informationstechnik • Institut für Automatisierungstechnik

MOBATSiM
Model-based Autonomous Traffic Simulation Framework

MathWorks®
MATLAB & Simulink
Simulink® 3D Animation™

MOBATSim: MOdel-Based Autonomous Traffic SIMulation Framework

Features:

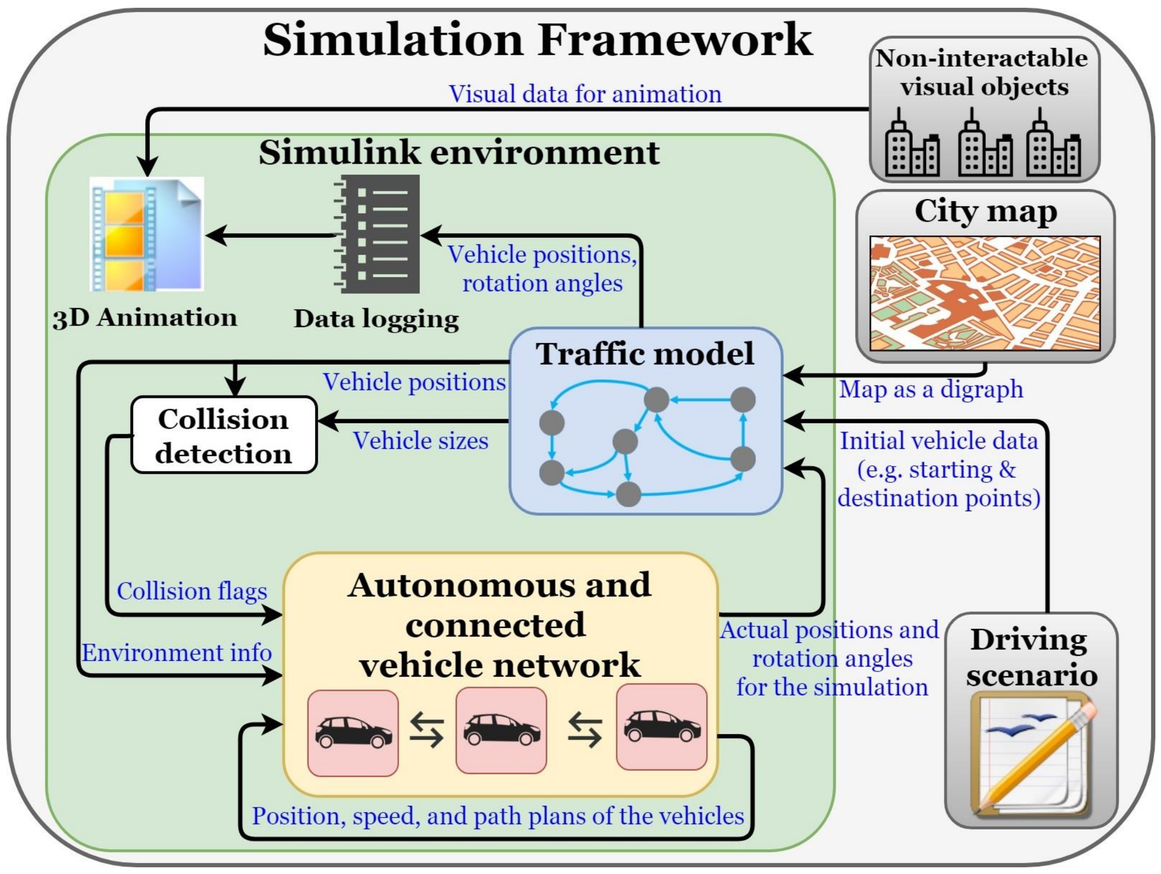
- Designed in Simulink as Model-based
- Flexible, reusable and easy to customize
- Animated by Simulink 3D Animation™

Goals:

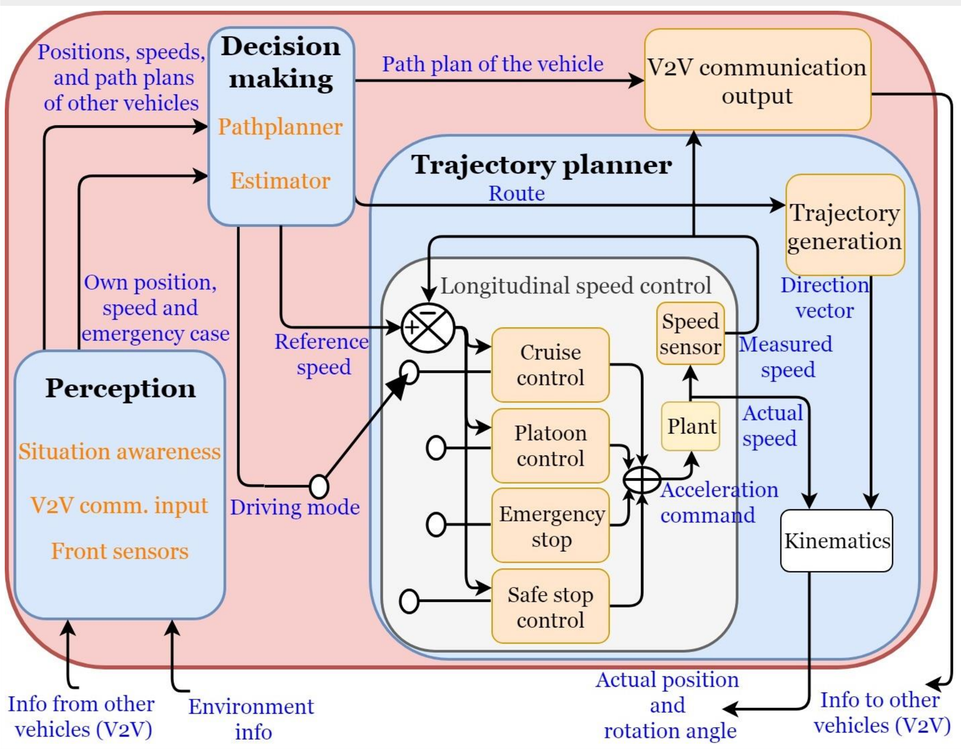
- **Safety analysis** of autonomous and connected vehicles and their functionalities (ADAS) in an urban traffic
- **Testing** control and decision algorithms of intelligent vehicles
- **Error propagation analysis** by scenario-based test case generation
- Validation and verification according to ISO 26262 using **simulation-based fault injection methods**



1. Architecture of MOBATSIM

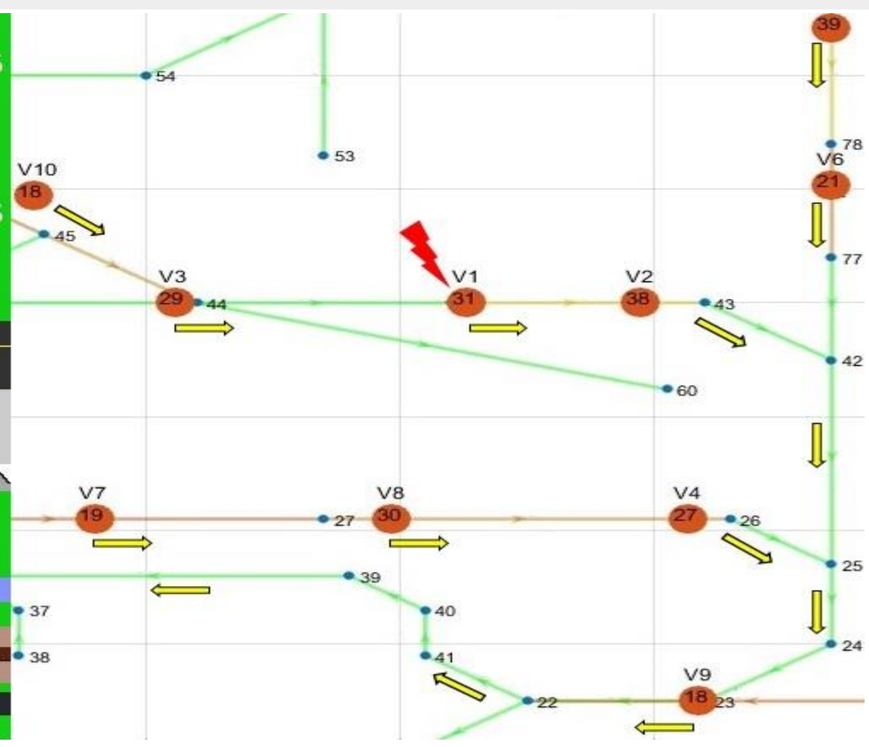


2. Autonomous Vehicle Structure

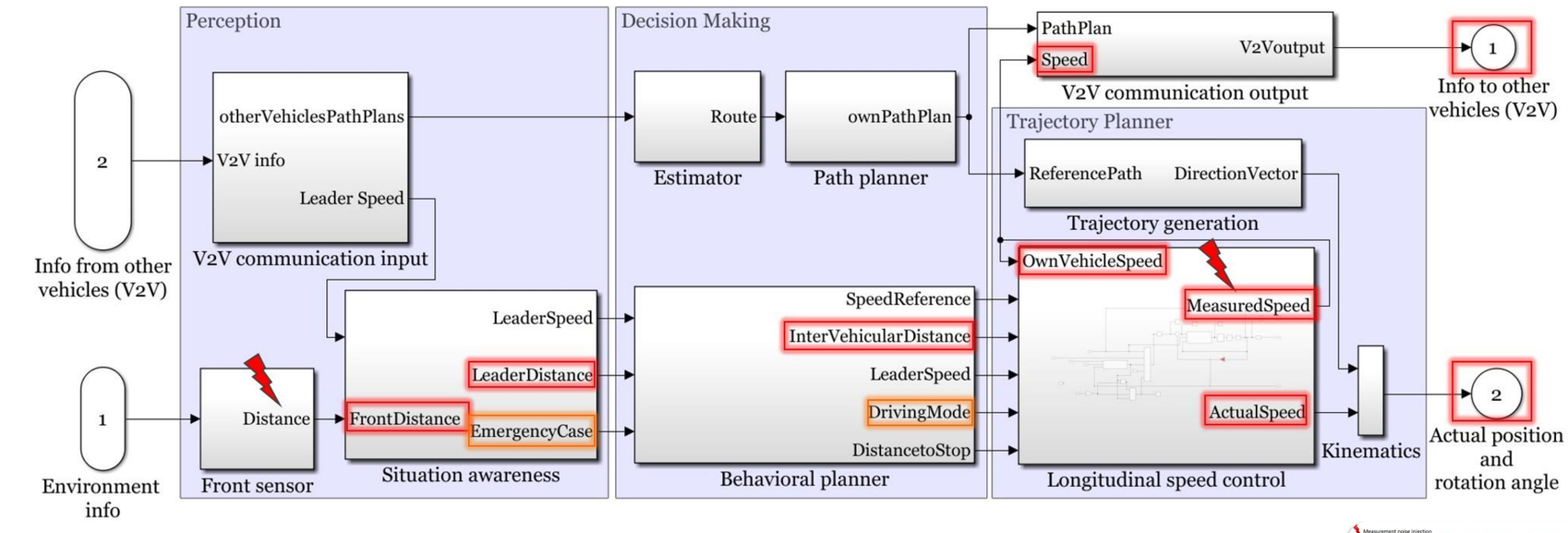


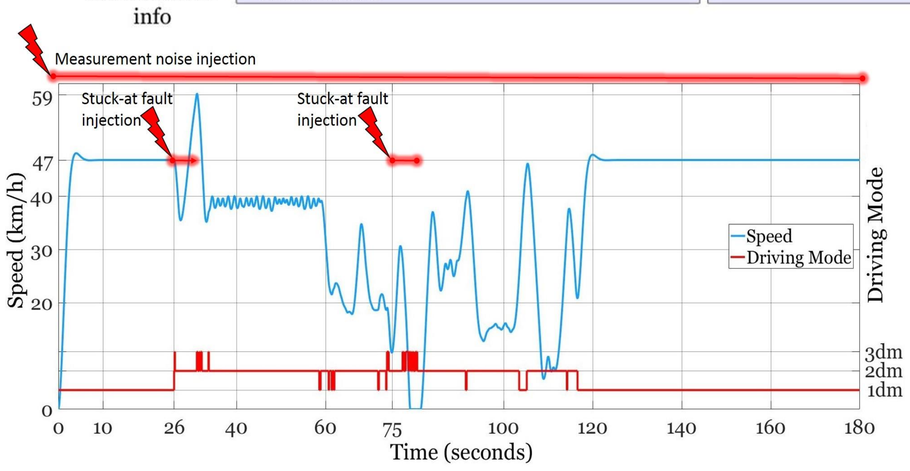
3. Scenario-based Fault Injection

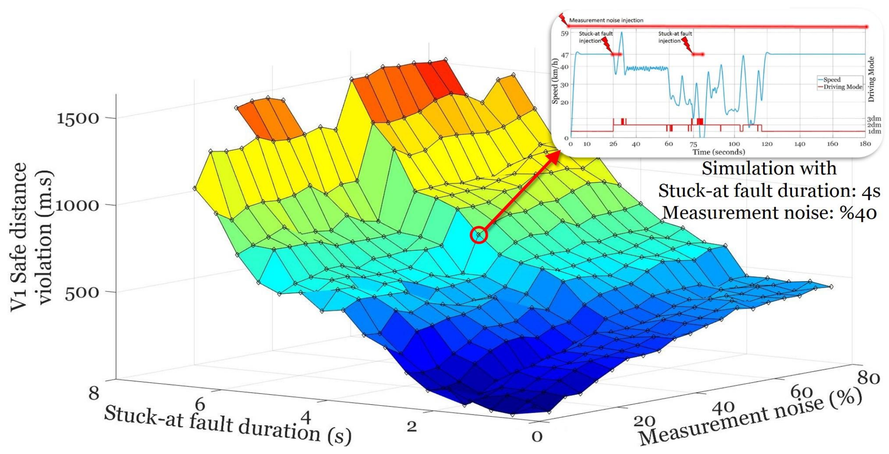




4. Error Propagation Analysis and Results







Publications and Awards

Saraoglu, M., Morozov, A. & Janschek, K. (Accepted paper), MOBATSim: MOdel-Based Autonomous Traffic Simulation Framework for Fault-Error-Failure Chain Analysis, IAV2019, Poland.

Hart, F., Saraoglu, M., Morozov, A. & Janschek, K. (Accepted paper), Fail-safe Priority-based Approach for Autonomous Intersection Management, IAV2019.

❖ **Simulink Student Challenge 2018 - 1st Place**



Saraoglu, M., Morozov, A. & Janschek, K. (2019, In Press). Safety Assessment of Autonomous and Connected Vehicles by a Model-based Traffic Simulator, 19th International Stuttgart Automotive and Engine Technology, Stuttgart.

Saraoglu, M., Hart, F., Morozov, A. & Janschek, K. (2018, In Press). Fault-Tolerant Path Planning in Networked Vehicle Systems in Presence of Communication Faults, NecSys 2018 7th IFAC Workshop on Distributed Estimation and Control in Networked Systems, Groningen, The Netherlands.

Saraoglu, M., Morozov, A., Söylemez, M. T. & Janschek, K. (2017). ErrorSim: a Lightweight Tool for Error Analysis of Simulink Models, International Conference on Computer Safety, Reliability and Security, Trento, Italy.

MathWorks®
MINTec®
Rupprecht-Gymnasium München
Naturwissenschaftlichtechnologisches
und Sprachliches Gymnasium
ZUSAMMEN.WACHSEN.

Comparison of 4 Different Routing algorithms in MATLAB

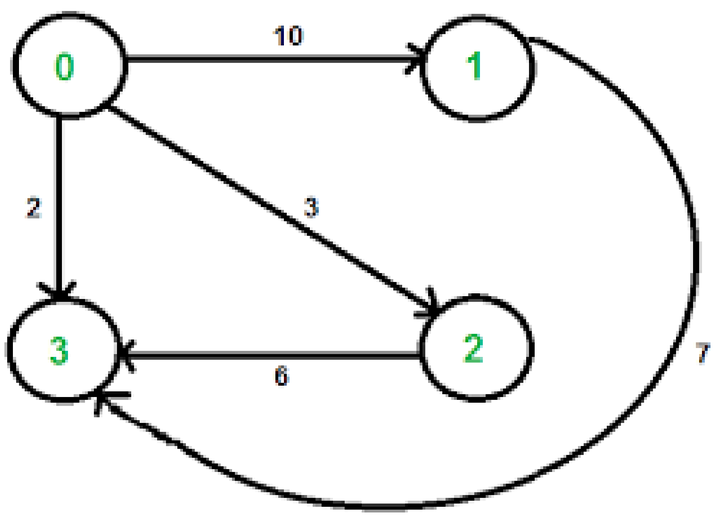
Basic Graph Theory:

A graph is an abstract data type. It has vertices (V) which are connected via edges (E). So the mathematical definition is: G=(V,E)

There are different types of graphs:

1. A graph can be weighted, which means that each edge (E) has a weight which can be seen as the length of the edge
2. A graph can be undirected or directed

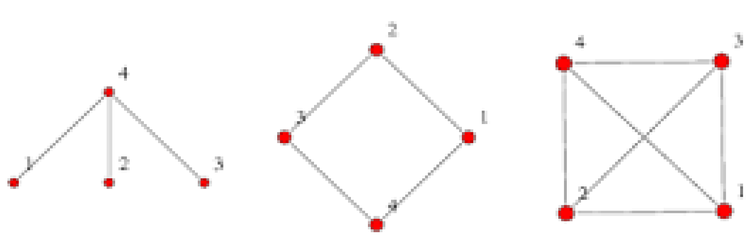
- The edges of an undirected graph do not have directions, which basically means that if a vertex v1 is connected to a another vertex v2, v2 is also connected to v1
- In case of a directed graph the edges do have directions, i.e. if vertex v1 is connected to vertex v2, it does not have to be that v2 is connected to v1



(example of a directed weighted graph)

Adjacency matrix (A):

- It is a |V| * |V| matrix where v are the vertices. Its cells represent the edge weights between the vertices i, j if A_(ij)∈A
- if i=j→A_(ij)=0 or if there is no edge between i,j→ A_ij=0



$$\begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

(adjacency matrices and the corresponding graph)

Incidence matrix (B):

That's a |v| * |E| matrix where v are the vertices and E are the edges

- f B_(i,j)= -1 edge j leaves vertex i
- If B_(i,j)=1 if edge j enters vertex i
- f B_(i,j)=0 if there is no connection

Algorithms:

Shortest path problem

This is the problem where you are given a graph G=(V,E) and two vertices, a start vertex and a target vertex. The aim is to find the shortest path to get from the start to the target vertex.

Bellman-Ford Algorithm

This algorithm tries to solve the shortest path problem. It computes the shortest path from 1 vertex to all vertices of the graph. It visits all the vertices in the graph V-1 times (V = amount of vertices in the graph) so that it now knows all the shortest paths starting from the start vertex. This algorithm can handle negative weights provided that there are no negative cycles; if so, it can at least detect these negative cycles.

Dijkstra

Dijkstra solves the shortest path problem. This algorithm visits each vertex once and computes the shortest path from the start vertex to all of the vertices.

A*

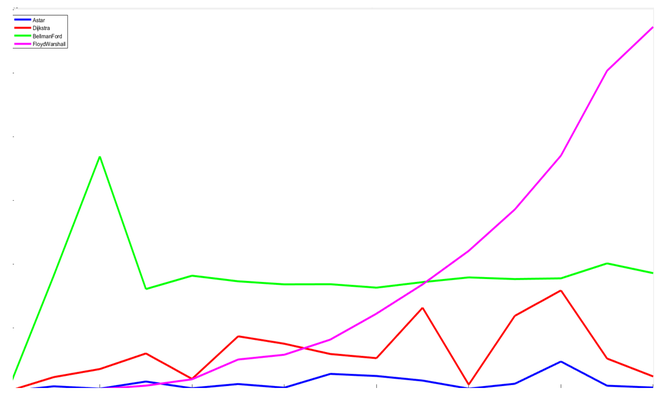
A* is similar to Dijkstra's algorithm but it uses heuristics. Basically Dijkstra only considers the weights of the edges. A* also considers the geographical position of the vertex. So i.e. it also looks at the beeline between the current vertex and the target vertex.

A* is the most efficient since you do not have to visit each vertex.

Floyd Warshall

This algorithm is used to solve the all-pairs shortest path problem. It finds for all vertices the shortest path to any other vertex in the graph. This algorithm can also be used to solve the shortest path problem by simply saving the predecessor of each vertex in order to be able to reconstruct the path. This algorithm should be the slowest since it is originally solving another problem.

Implementation:



(Pic. 3; x-axis = amount of vertices, y-axis = calculation time; blue: A*, red: Dijkstra, green: Bellman Ford, purple: Floyd Warshall)

Floyd-Warshall is the slowest, since this algorithm was not made for the shortest path problem. Anyway in case the graph has less then 100 vertices this algorithm is a decent algorithm.

A* is the fastest algorithm, since it does not have to visit all the Vertices, but it can still guarantee the best route.

Bellman Ford is slower than Dijkstra, because Bellman Ford has to visit each vertex several times, whereas Dijkstra only has to visit each vertex once.

ZUSAMMEN.WACHSEN.
MathWorks®
MathWorks
Adalperostr. 45, 85737 Ismaning
www.mathworks.de
MINTec®
MINT-EC
Poststraße 4/5, 10178 Berlin
info@mint-ec.de

University of Stuttgart

Helikopter - Flugsimulation

Maharjan, Rijan
Rothaupt, Benjamin
Walter, Gajana
Institut für Flugmechanik und Flugregelung (iFR)
Pfaffenwaldring 27, 70569 Stuttgart



Ultraleichthubschrauber

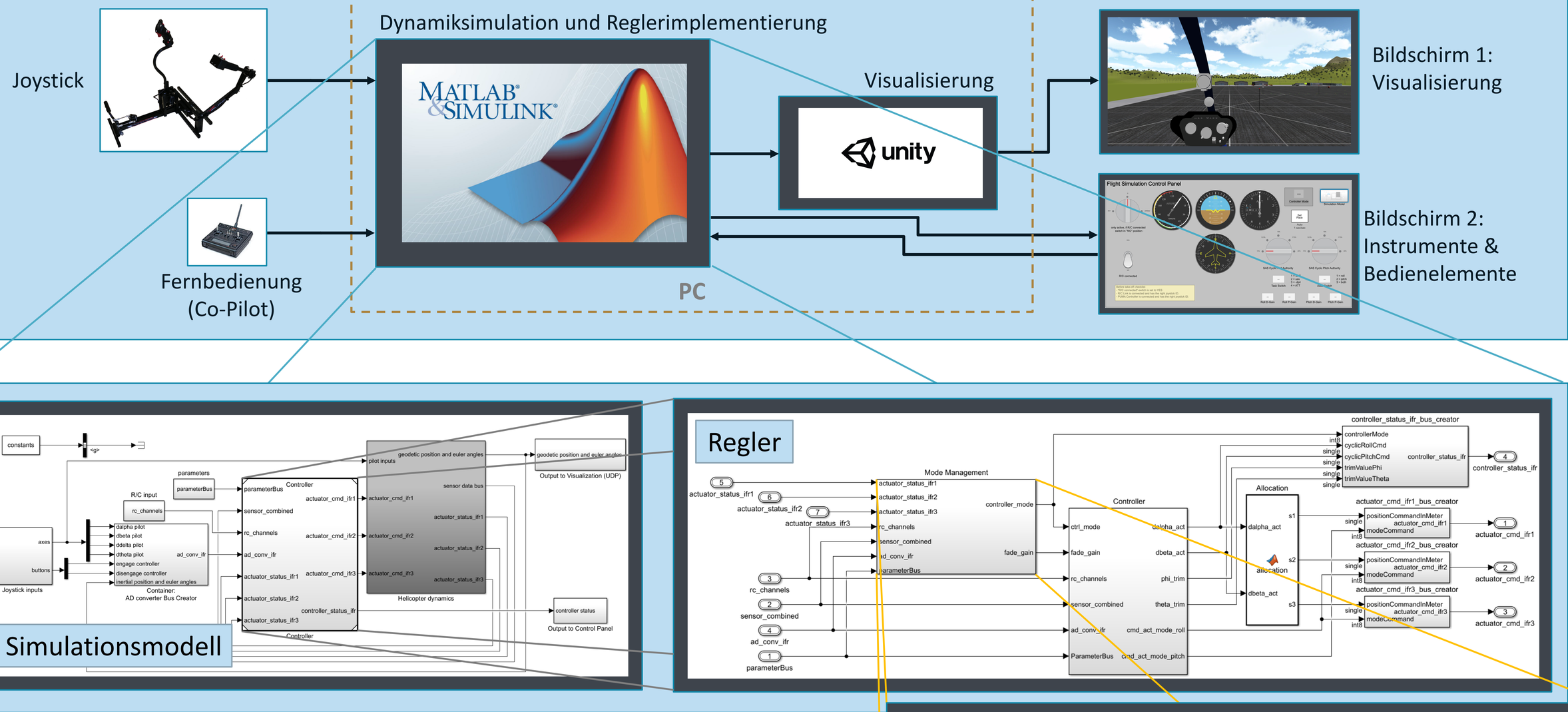
- Seit 2016 existiert in Deutschland die Ultraleichthubschrauber-Klasse
- Fluggerät und Pilotenlizenz sind vergleichsweise kostengünstig
- Eine Reduktion des Workloads kann – gerade bei Hobbypiloten – die Flugsicherheit erhöhen
- Hierfür eignet sich ein Regelungs- oder Stabilisierungssystem

Flugsimulation

Wozu braucht man hier eine Flugsimulation?

- Erprobung von Regelgesetzen
- Test von Reglerparametern vor dem realen Flug
- Pilotentraining
- Handling Qualities Evaluation

Bei der Flugsimulation kommt MATLAB & SIMULINK zum Einsatz.



Die hier gezeigte Flugsimulation wird bei der Entwicklung von Regelgesetzen für einen manntragenden UL-Hubschrauber verwendet.

Der Inhalt des Regler-Subsystems wird nach erfolgreicher Erprobung in der Simulation automatisch in C-Code übersetzt und ist auf dem Bordrechner lauffähig.

iFR

edm aerotec GmbH

Hardware in the Loop Experimente für die Multicopter Entwicklung

Einsatz von MATLAB Simulink zur effizienten Entwicklung, Analyse und Optimierung von autonomen Multicoptern.

P. Schattovich, M. Hochrainer, Fachhochschule Wiener Neustadt, 2700 Wiener Neustadt, Austria

FACHHOCHSCHULE WIENER NEUSTADT
Austrian Network for Higher Education

Einleitung

Die HiL Methodik erlaubt die Reduktion von Entwicklungszeiten und das realistische Testen von Komponenten und Modulen. Dazu werden physikalisch verfügbare Komponenten (reales Modell) mit einem Computer-Simulationsmodell (virtuelles Modell) gekoppelt, welches die (noch) nicht vorhandenen Komponenten sowie alle wesentlichen Umgebungseinflüsse repräsentiert. Mit HiL sind während der gesamten Entwicklungs- und Nutzungsdauer aussagekräftige Tests über die Systemgrenzen hinaus möglich. Eine typische Anwendung stellt die Entwicklung eines Multicopter (siehe Abb. 1) HIL-Prüfstands und die darauf basierende Visualisierung und Optimierung des Flugverhaltens dar.



Abb. 1: Prototyp vom untersuchten Multicopter

Konzept - Umsetzung

Die Komplexität von HIL Experimenten wird von der Wahl der Systemgrenzen wesentlich beeinflusst wobei eine „digitale Schnittstelle“ einfacher zu realisieren ist als eine „physikalische Kopplung“, welche die Erfüllung von dynamischen Gleichgewichtsbeziehungen in der jeweiligen Domäne erfordert. Daher erfolgt die Trennung zwischen real vorhandenen Komponenten (Steuergerät, Motortreiber, Motoren, Fernsteuerung) und dem Multicopter – Simulationsmodell durch die Weiterleitung simulierter IMU (Inertial Measurement Unit) Sensorwerte an die Multicopter Steuereinheit (ECU). Der ECU-Lageregler berechnet daraus die erforderlichen Motorsteuersignale, und sendet diese zurück an das Simulationsmodell womit die HIL Schleife geschlossen wird, siehe Abb. 2.

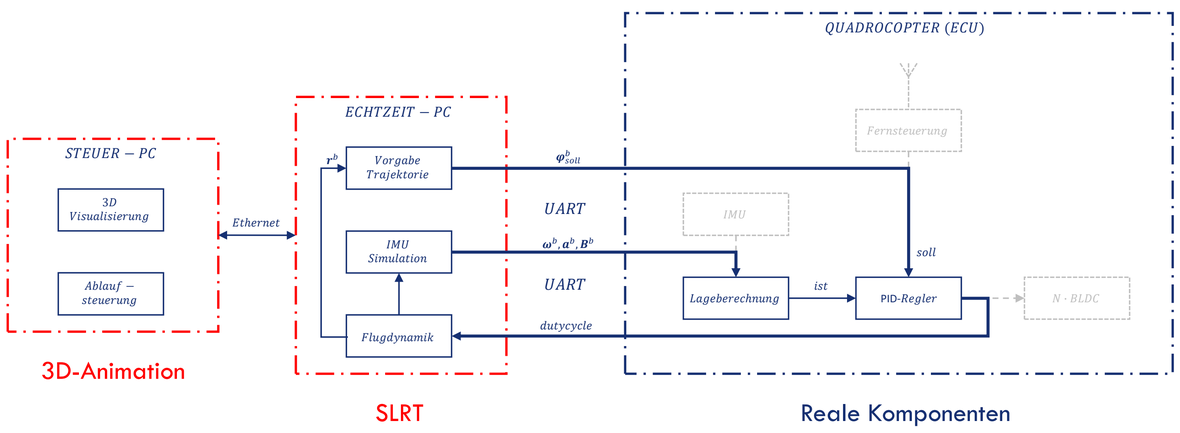
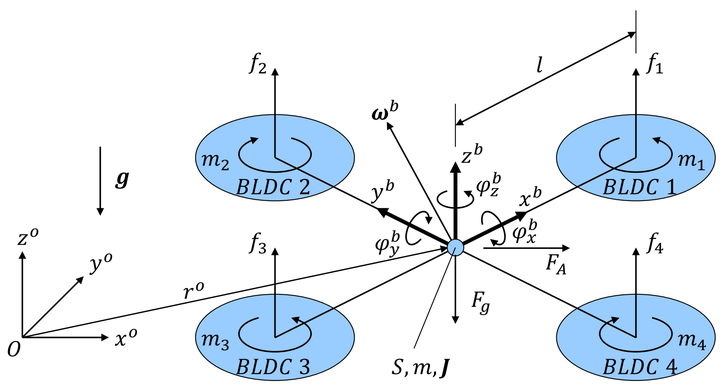


Abb. 2: HiL Prüfstand: Signalfluss im virtuellen und realen Modell

Die Multicopter Flugdynamik wird mit Hilfe von Starrkörpermodellen beschrieben, siehe Abb. 3. Die numerische Lösung der nichtlinearen Bewegungsgleichungen erfolgt in einem diskreten Simulink Modell mit einer konstanten Abtastzeit von $T_s = 5\text{ms}$. Das gesamte Simulationsmodell wird mit Hilfe von automatisierter Codegenerierung auf einen handelsüblichen PC (Simulink RealTime) übertragen und ausgeführt. Alle erforderlichen Systemparameter sind, soweit nicht direkt messbar, durch Laborversuche bestimmt.



$$m\dot{\boldsymbol{v}}^b + m\boldsymbol{\omega}^b \times \boldsymbol{v}^b = \sum \boldsymbol{F}_i^b + \boldsymbol{F}_g^b + \boldsymbol{F}_L^b$$

$$\boldsymbol{J}^b\dot{\boldsymbol{\omega}}^b + \boldsymbol{\omega}^b \times \boldsymbol{J}^b\boldsymbol{\omega}^b = \sum \boldsymbol{M}_i^b$$

Abb. 3: mechanisches Modell mit korrespondierenden Bewegungsgleichungen

Ergebnisse

Nach passender Kalibrierung und entsprechenden Funktionstests ist der HIL-Prüfstand für unterschiedlichste Experimente einsetzbar. Etwa die Evaluierung von Sensor- und Filtereigenschaften, die Abschätzung des Regelverhaltens oder die Simulation komplexer Flugmanöver.
Erst durch den Einsatz von jederzeit reproduzierbaren HIL Experimenten können Schwachstellen im Zusammenspiel von Sensoren, Aktoren und der Steuereinheit identifiziert und behoben werden: wenig Stabilitätsreserven im Schwebeflug, seltene Abstürze aufgrund von Motorausfällen sowie Firmware Mängel in der Sensorauswertung. Abb. 4a zeigt die Verbesserung der Multicopter-Dynamik bei einen Gierwinkelsprung von 10°. Um das simulierte Flugverhalten zu beurteilen, wurde eine 3D Visualisierung entwickelt, siehe Abb 4b.

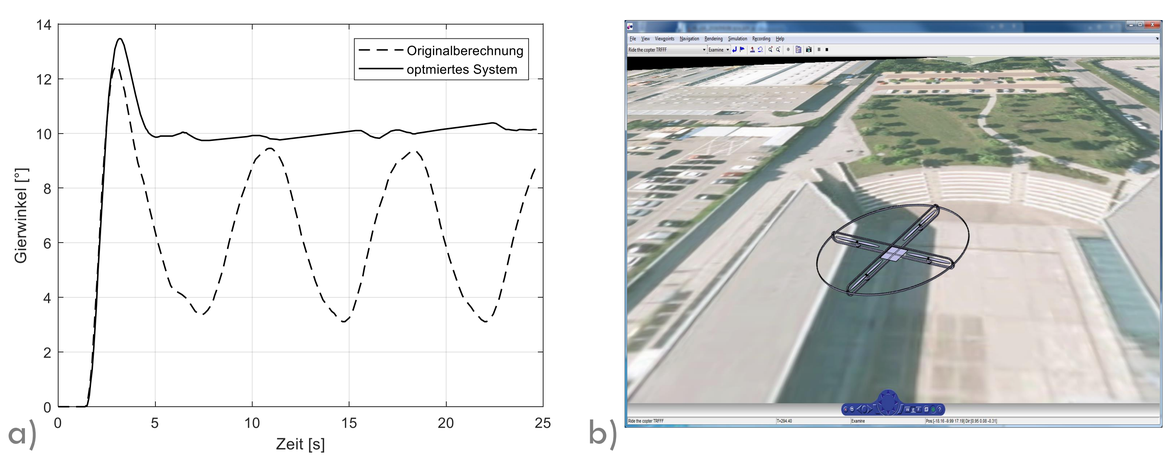


Abb. 4: a) Systemverhalten bei Gierwinkelsprüngen vor bzw. nach Optimierung; b) Visualisierung des Flugverhaltens in Simulink 3D-Animation

Nach Verifizierung der HIL-Ergebnisse durch reale Testflüge kann der entwickelte HIL-Prüfstand auch für komplexere Aufgaben wie etwa dem Testen von autonomen Multicoptern Verwendung finden. Dabei ist die Möglichkeit der Definition bzw. Änderung sämtlicher Randbedingungen besonders hilfreich: simulierte GPS Daten mit statistischem Fehler können ebenso berücksichtigt werden wie Windeinflüsse, Gewichtsänderung, Ladezustand der Batterie, variable Nutzlasten oder Fehler/Störungen in den Sensorsystemen. Abb. 5 zeigt die Ergebnisse eines virtuellen HIL Fluges mit vorgegebener Trajektorie auf Basis von simulierten GPS Daten.

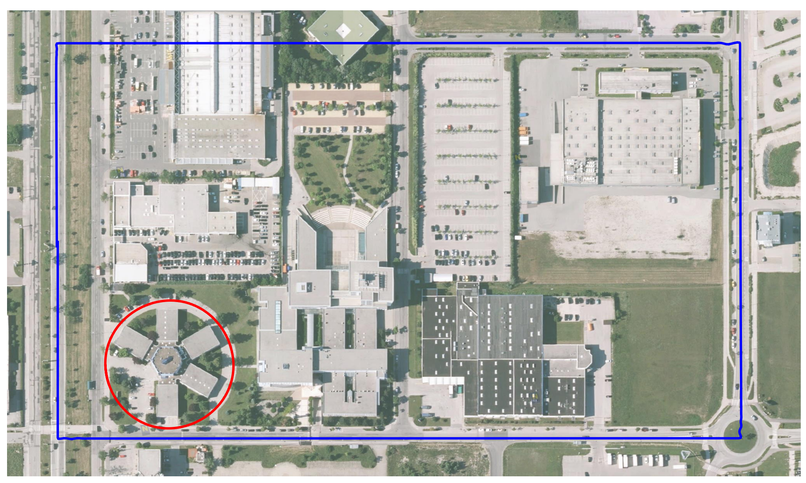


Abb. 5: Flugtrajektorien bei autonomen HiL Flug

Diskussion

Der vorgestellte Multicopter HIL-Prüfstand ermöglicht eine jederzeit reproduzierbare Durchführung und Analyse von beliebigen Flugmanövern unter äußerst realistischen Bedingungen. Somit ist eine Optimierung einzelner Komponenten möglich, bevor das Gesamtsystem verfügbar ist. In der gezeigten Anwendung kombiniert die HiL Philosophie alle Vorteile des experimentellen Testens mit jenen einer reinen Simulation.

Literatur

Ronald Stärz, 0x0 – 0xF from scratch to autonomous , Konferenz der Mechatornik Plattform, 22.11.2018, Graz
Markus Hochrainer, Peter Schattovich, Real-Time Hybrid Simulation of an Unmanned Aerial Vehicle, IMAC XXXV, 30.1-2.2.2017, Garden Grove, California, USA

The Hohmann - Transfer

Eva Röckl

Introduction

Already 100 years before the first flight to the moon, the author Jules Verne described in his books "From the Earth to the Moon", a gunnery club that built a cannon out of boredom at the time of the American Civil War to use it to send a bullet to the moon. [2]



These novels have fascinated the German space pioneer Walter Hohmann, who invented the so-called "Hohmann transfer" in 1925 with the help of Ziolkowski's 1911 recordings.

To this day, this transfer option is regarded as the most energetically favourable transition between two orbits around a dominant celestial body.

However, this only applies if both lanes between the Hohmann Railway are coplanar and circular.

Furthermore, their radii must have a ratio below 11.94. [3]

Description of the Hohmann transfer

The „Hohmann transfer“ describes the possibility of the level crossing of a satellite or a rocket by speed change.

There are two acceleration phases.

The so-called Kick1, the circular, near-Earth orbit changes into an elliptical, with the previously circled celestial body is located on one of the focal points of the ellipse.

At the point where the satellite has reached the apogee, it comes to a second acceleration phase - also called Kick2. The elliptical orbit is transformed back into a circular orbit.

However, its radius is much larger than that of the first one.

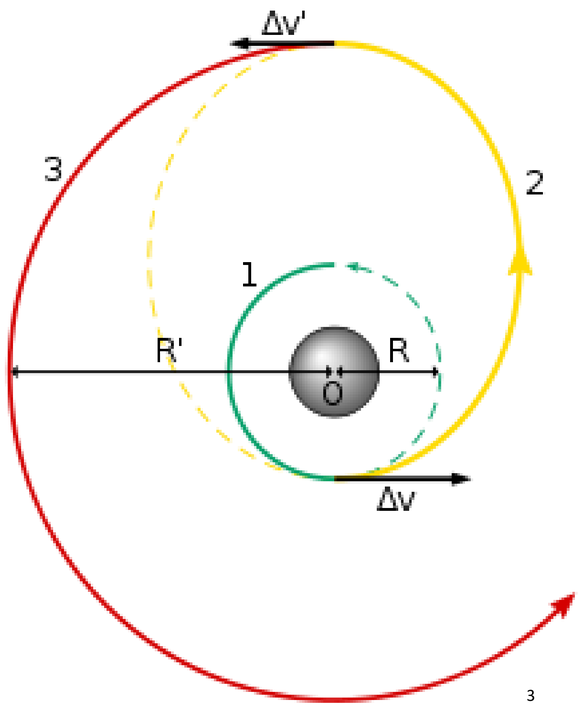
This elliptical transfer orbit between the near-Earth and orbital orbit is also called the Hohmann orbit and runs tangentially to both.

This transfer is seen as energetically favourable.

However, this only applies if both lanes between the Hohmann Railway are coplanar and circular.

Furthermore, their radii must have a ratio below 11.94.

If this is not the case, the so-called "bi-elliptic transfer" is energetically more favourable. [4]

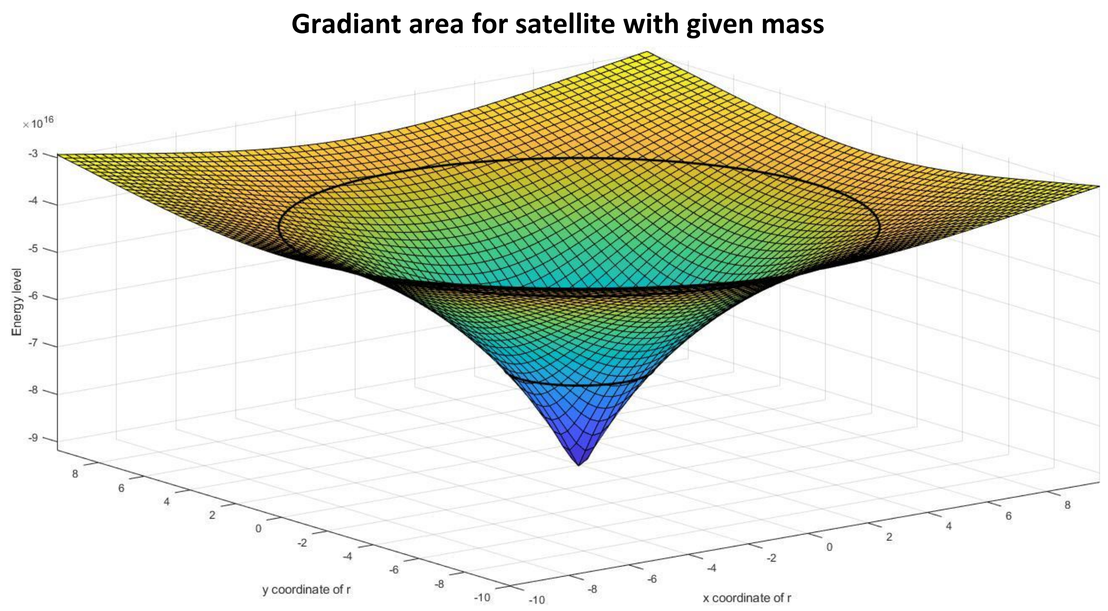


3

Implementation with MATLAB

The aim of the work is to prove that the „Hohmann transfer“ is the most energetically favourable of all previously known methods, under the conditions already mentioned.

For this purpose, a potential funnel is created in MATLAB to prove analytically that it is possible to use the direct path between two radii.



Future Implementations

If the potential funnel has been created and the analytical proof provided, it must be compared with other transfer possibilities. On the one hand the swing-by-maneuver will be examined and implemented and on the other hand the bi-elliptic transfer.

To provide this comparison, various conditions are created, such as different dominating celestial bodies, satellites of different masses and changing radii ratios.

The comparison with other transfer options is an important part of the work to show that the „Hohmann transfer“ is not just energetically favourable, but actually the most energetically favourable.

1 https://www.whs-hardheim.de/der-schulverbund/walter-hohmann/
2 http://www.j-verne.de/verne6.html
3 https://de.wikipedia.org/wiki/Hohmann-Transfer
4 https://de.wikipedia.org/wiki/Hohmann-Transfer
https://physik.cosmos-indirekt.de/Physik-Schule/Hohmann-Transfer
http://www.mabo-physik.de/hohmantransfer_info.pdf

Rupprecht-Gymnasium München
Albrechtstraße 7
80636 München
mint@rupprecht-gymnasium.de

MathWorks
Adalperostr. 45, 85737 Ismaning
www.mathworks.de

MINT-EC
Poststraße 4/5, 10178 Berlin
info@mint-ec.de

BACnet stack integration with BACnet4Simulink

Christian Bock, Alexander Martens, Olaf Simanski, Olaf Hagendorf

Hochschule Wismar - University of Applied Sciences: Technology, Business and Design
Research Group Computational Engineering and Automation (CEA)
www.cea-wismar.de

Contact: christian.bock@hs-wismar.de

Lüftung
Klima & Kälte
Reinraumtechnik

EUROPÄISCHE UNION
Europäischer Fonds für
Regionale Entwicklung

Mecklenburg Vorpommern
MV tut gut.

Abstract

As part of a R&D (research and development) project of the University of Wismar, a modified open-source BACnet™ (Building Automation and Control Networks) library is used as a microcontroller protocol stack for device-to-device communication. As another result of this, a Simulink® block set utilizing BACnet/IP as communication protocol was developed. Target of this port is the interaction of a Simulink® models using BACnet4Simulink blocks with multiple BACnet enabled devices. BACnet as communication protocol provides all necessary prerequisites for a vendor and application independent device-to-device communication alongside a common interface, such as IP based Ethernet networks. BACnet4Simulink implements the main BACnet functionality to interact with BACnet/IP devices in the form of individual Simulink® blocks.

Project backed by:
TBI MV, support code: TBI-V-1-135-VBW-048

BACnet Protocol

BACnet, originally developed by ASHRAE (American Society of Heating, Refrigerating and Air-Conditioning engineers) in 1987, is defined as a standard by the ANSI/ASHRAE 135 (since 1995) and ISO 16484-5 (since 2003) norms. It is a protocol for building automation purposes at process control level. Apart from its origin as process control protocol, BACnet is fully capable of implementing a device-to-device communication network. Therefore, BACnet supports to run a complete automation network in the sense of individual devices, or “things”, directly communicating with each other. This capability enables the implementation of BACnet as a means of communication for smart devices for IoT (Internet of Things) purposes. The underlying principle of BACnet is to ensure vendor independent interoperability between devices. This is achieved by the definition of a standardized representation as a sum of objects and services, detailed by object properties. Figure 1 shows a general representation of a generalized BACnet protocol schematic.

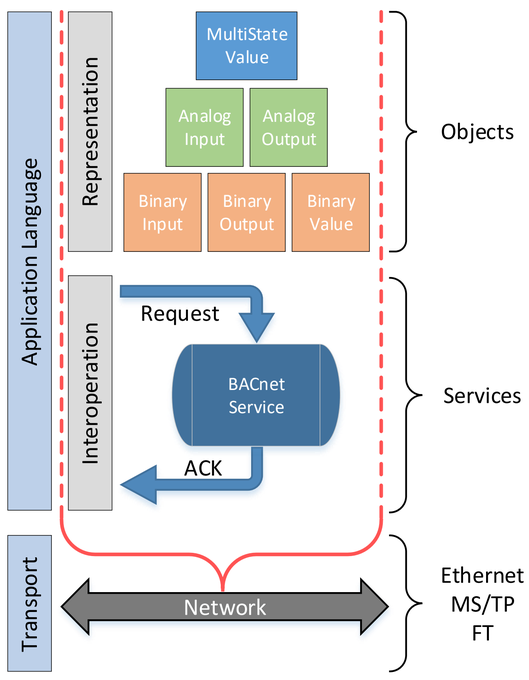


Figure 1: BACnet schematic

Model - Cleanroom with Air-lock chamber

Figure 2 shows a real world model of a physical clean room environment used to simulate the influence of changes in airflow volume on a single chambers pressure conditions. The pressure measurements and the control of in- and outlet fans as well as the airflow measurements are controlled and supervised by STM32F746ZG controllers running a BACnet stack. The room pressures inside the models chambers, the cleanroom and the air-lock chamber, are being controlled by a Simulink® model based on BACnet4Simulink running on a PC. An example implementation of such a feedback-control model can be seen in Figure 6.

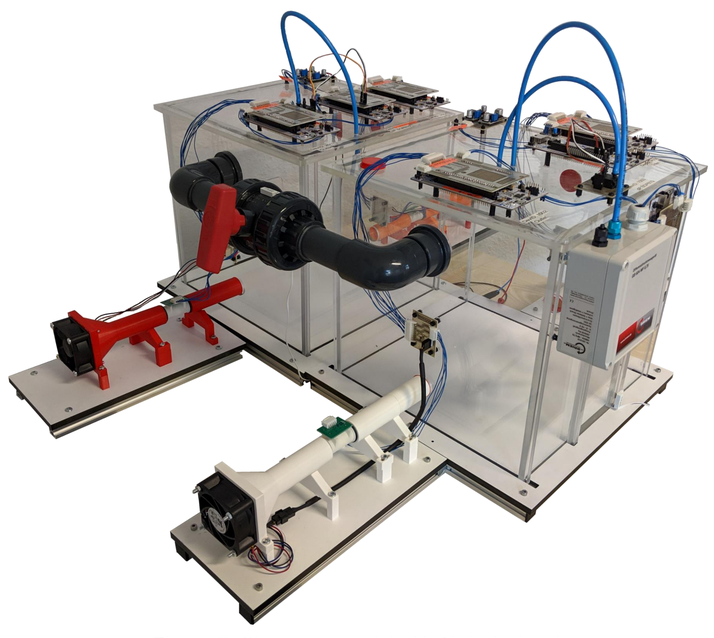


Figure 2: Cleanroom model with Air-lock chamber

BACnet - Stack performance

As a performance evaluation of BACnet4Simulink, a BACnet4Simulink model is communicationg with a BACnet device running on a STM32F7 microcontroller. The model polled 5 *AnalogInput* objects, hosted by the controllers BACnet stack, at a rate of 20ms. The average response time across all five objects averages at roughly 7ms. The maximum response time was measured with 25ms.

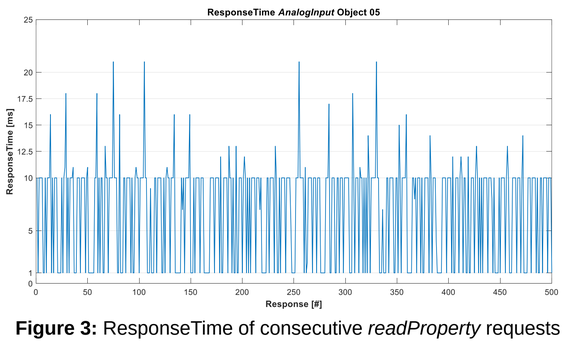


Figure 3: ResponseTime of consecutive *readProperty* requests

BACnet-Stack block set

The integration of a BACnet stack into Simulink® is done by defining a single MATLAB Level 2 s-function. The same S-function is used for all four Simulink® BACnet/IP blocks but with different functionality, depending on arguments passed during block invocation.

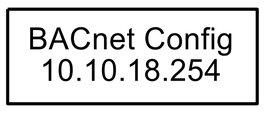


Figure 5: BACnet4Simulink Config block

The BACnet Config Block acts as the invocation instance of the underlying BACnet stack itself. During it's invocation, the stack collects and reacts to received BACnet APDU (Application Protocol Data Unit) messages.

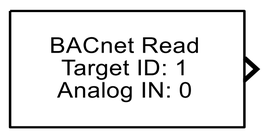


Figure 6: BACnet4Simulink Read block

For the retrieval of an objects property value, the BACnet4Simulink *Read* block is used. It issues a *readProperty* request and waits for an answer, marked with an *invokeID* identical to the ID issued during the request.



Figure 7: BACnet4Simulink Write block

Similar to the BACnet4Simulink Read block, the Write block issues a *serviceRequest* marked by an *invokeID*. In this case, the to be invoked service, modifies an objects property. The answer to this request is an acknowledgement marked by the *invokeID* of the preceding *serviceRequest*.

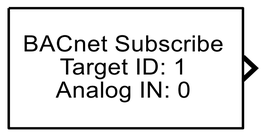


Figure 8: BACnet4Simulink Subscribe block

The BACnet4Simulink Subscribe block represents the capability of BACnet to utilize event-based communications. The subscribe block, is used to notify a BACnet device about a subscriber requesting to be notified in the case of a property being modified past a certain threshold. Equivalent to the Read and Write block issuing an *invokeID*, the Subscribe block, issues a *processID* to assign subscription-notifications to the actual subscriber.

Example - BACnet - Simulink®

With BACnet4Simulink, Simulink® models, can integrate BACnet enabled sensors and actors. Figure 6 shows two discrete PI controllers, regulating the RPM of a Fan via PWM signal in accordance to the fans period time. Each such Simulink® model has to be comprised of at least the config block 'BACnet Config', detailing the interface to be used for BACnet communication. The Simulink® Real-Time Sync block is optional, but provides the means to execute individual blocks at a given frequency without further ado.

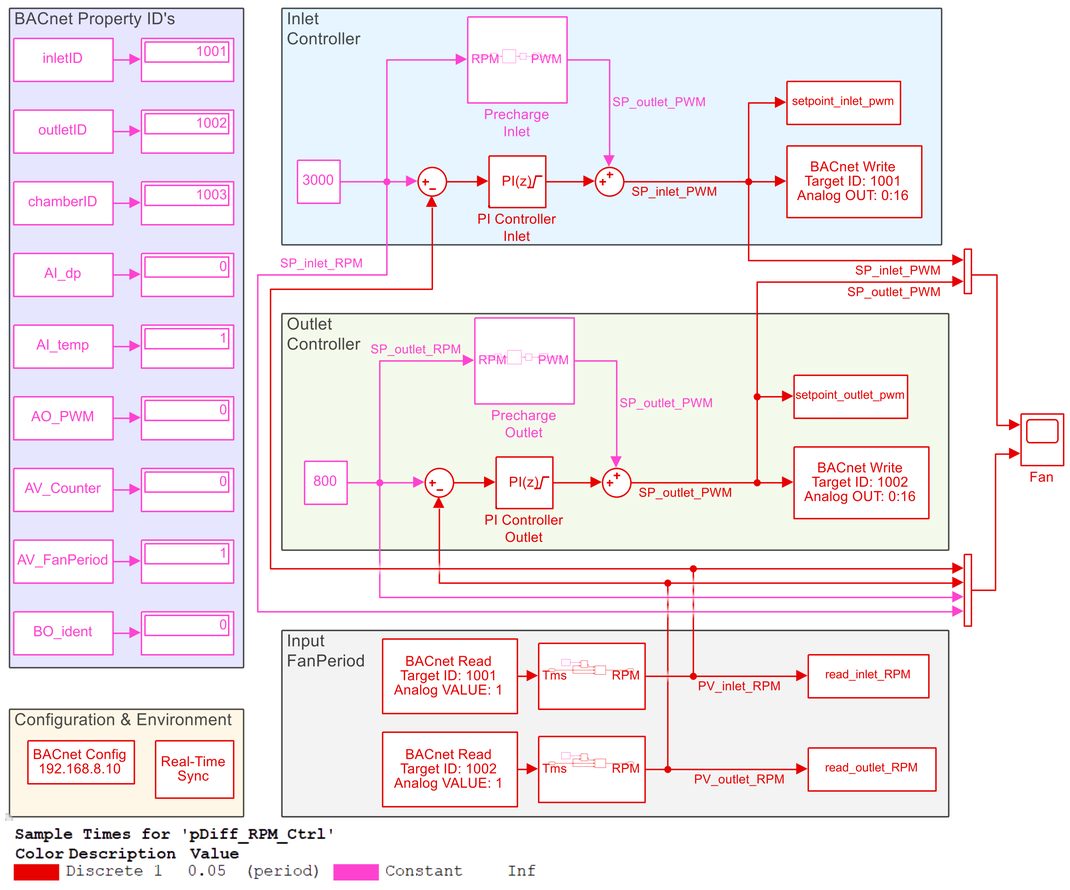


Figure 6: PWM Fan - RPM controlled with BACnet4Simulink

Conclusion

The flexibility of BACnet to project virtually any source of information as a group of standardized objects and services allows it to integrate a wide range of devices into IP based Ethernet networks via a common network interface. Therefore, the BACnet4Simulink library effectively extends the feature set of Simulink® with device-to-device communication capabilities for BACnet/IP enabled devices. The cleanroom model, as pictured in Figure 2, depicts an example of how a single Simulink® model, utilizing Simulink4BACnet blocks, interacts with and controls numerous independent devices. In this case, the Simulink4BACnet block set is used to evaluate control algorithms of real world clean rooms utilizing the vast possibilities of a Simulink® driven control design.

GitHub:
https://github.com/ATM-HSW/BACnet4Simulink

BACnet® is a registered trademark of ASHRAE Simulink® is a registered trademark of The MathWorks, Inc.

Mbed Target and IOT – A ThingSpeak Simulink® Block Set

Olaf Hagendorf

Hochschule Wismar - University of Applied Sciences: Technology, Business and Design
Research Group Computational Engineering and Automation (CEA)
www.cea-wismar.de

Contact: olaf.hagendorf@hs-wismar.de

Abstract

MATLAB/Simulink is widely used in engineering applications, among others in robotics, control systems, information technologies etc.
Simulink as a widely used data flow oriented development tool, can also be used to program MCUs. Beside the Mbed Target, introduced with [1], several other targets exist. They are provided by Mathworks or are available as third-party tools. Examples of these are e.g. [2], [3]. Whereas the provided functionality of the first package is very basic, the second one offers nearly as much functions as the MCU hardware provides. But all variants except Mbed Target are closed source and can't be extended because of new needs.
Since a couple of versions MATLAB supports the IOT platform ThingSpeak with local interfaces to read and write as well as possibilities to evaluate the gathered data directly on the cloud servers.
With this publication an extension of Mbed target is introduced to combine the code generation of Simulink, MATLAB and Embedded Coder and an IOT interface to ThingSpeak. It enables the usage of MATLAB/Simulink at both ends of the communication line, at the edge and in the cloud.

Mbed OS - Software

Mbed OS is an open source, object oriented library for Cortex-M micro-controllers. Various Python-based tools, a variety of configuration options etc. are completing the software system. The historic idea behind Mbed OS is similar to the Arduino ecosystem, making the start of microcontroller programming as easy as possible.

Figure 1 shows a simplified structure of the library. Whereas the driver part is comparable to Arduino, a huge amount of extensions e.g. RTOS, files systems, security and networks libraries are part of it. The library was introduced by ARM and is developed as well as by professional developer groups and by the community. The project is hosted at Github.

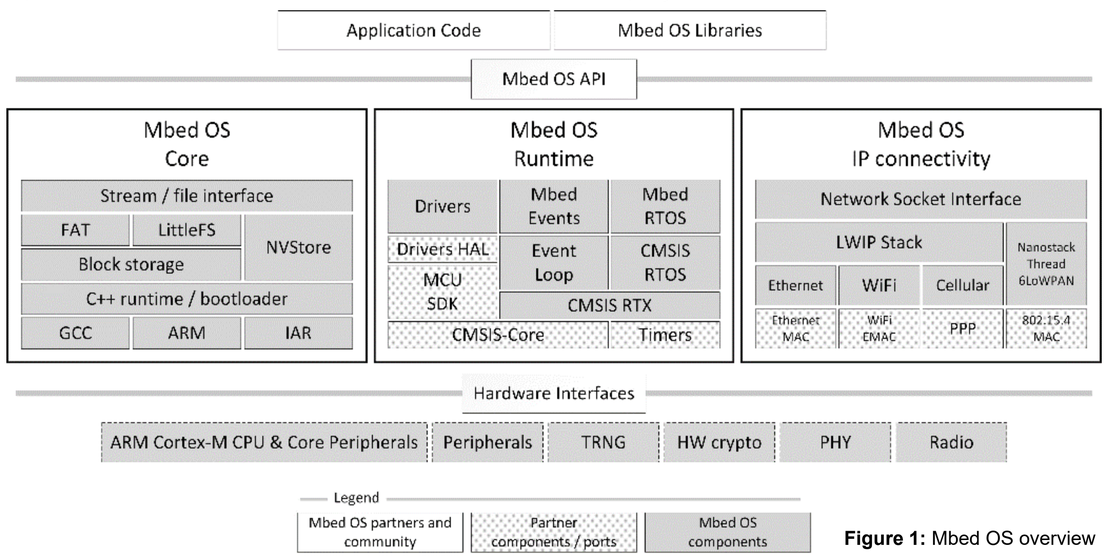


Figure 1: Mbed OS overview

Mbed OS - Hardware

Mbed OS is a portable library, whereas the application programming interface is C++ based but the underlying adaption layer implies plain C. The bases are company specific hardware abstraction layers e.g. STM32Cubexx by STM [ref]. Between these HALs and the lower Mbed OS layer only a thin adaption is necessary. This principle allows an easy integration of new hardware platforms.
There exists a huge amount of Mbed OS compatible hardware, 306 boards by 14 chip manufactures at the time of writing. A Mbed OS compatible board has to have an USB debug adapter with 3 functions: (i) a debug function like DAP-link, ST Link or similar, (ii) an USB-MSD mass storage interface i.e. a USB stick function and (iii) a USB-CDC serial interface. Except of the debug interface, no additional drivers are necessary to program an Mbed compatible board and communicate with it. Common form factors for Mbed compatible boards are Arduino Uno and Arduino Nano so hardware extensions e.g. shields and base boards of the Arduino ecosystems can be used. Other form factors exists as well.
Figure 2 shows three board variants of the Nucleo and Discovery families produced by STM [4].

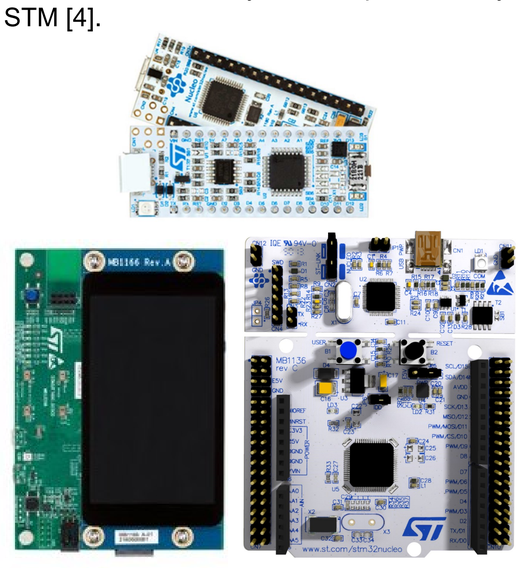


Figure 2: Three Mbed compatible boards by STMicro

Mbed Target

Mbed Target is a coder support packages. It is completely open from both sides: all code and configuration files i.e. MATLAB code, templates, configurations etc. as well as all underlying MCU libraries are available free of charge and as source codes. It is based on Mbed OS 5, therefore all Mbed OS 5 compatible microcontrollers can be used to run Simulink models. Mbed Target uses the open source GNU GCC to compile the generated source code, whereas project files for all Mbed OS supported IDEs can be additionally generated.
Figure 3 depicts the principle workflow of the code generation within Mbed Target. After generation of the model specific c/cpp files, Mbed OS tools are used the generate suitable makefiles for the following compile/link/flash part. The automatic makefile generation of Mbed OS is used but was adapted to the needs of the coder toolboxes.
MbedTarget is compatible with Embedded Coder as well as Simulink Coder.

Simulink® is a registered trademark of The MathWorks, Inc.

Mbed Target

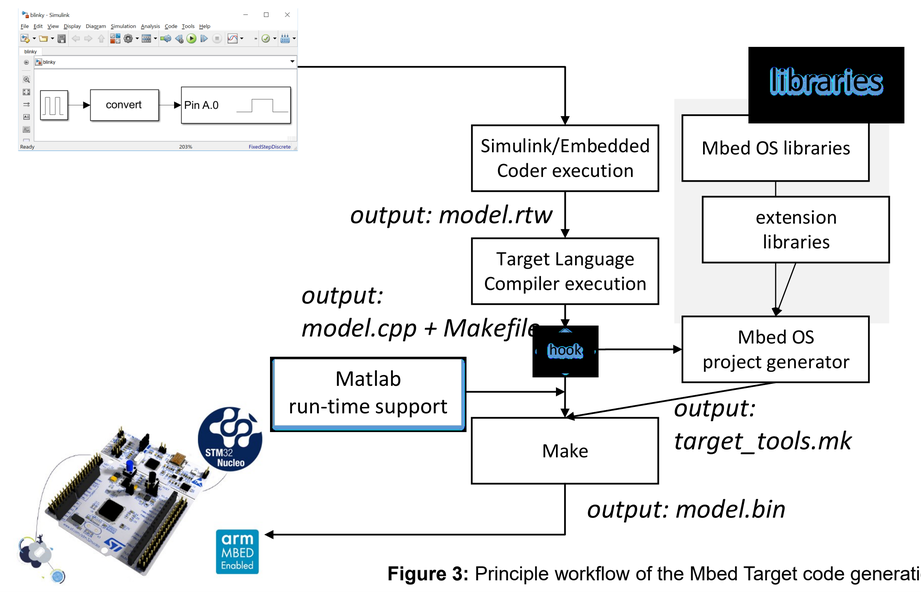


Figure 3: Principle workflow of the Mbed Target code generation

ThingSpeak Block Set

ThingSpeak offers two communication protocols: http and MQTT. Both provides the possibility of setting or getting all/some fields of a channel or only one specific field. The introduced block set offers blocks for all variants: (i) http read and write blocks for a complete channel or a part of it, (ii) MQTT publish and subscribe blocks for a specific field of a channel. All 4 blocks reads the configuration directly from ThingSpeak, only datatypes has to be included. Figure 4 shows the 4 blocks together with a specific MQTT configuration block. The block set expects an IP stacks which is already available within Mbed Target. Figure 5 depicts three config masks.
The underlying basis of the ThingSpeak block set, i.e. for the http part, is the Mbed library ThingSpeak4mbed [5]. This library is a port of the Arduino ThingSpeak library provided by Mathworks.

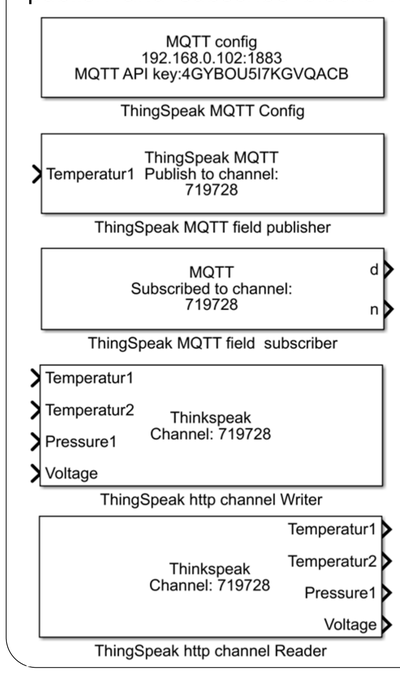


Figure 4: ThingSpeak block set

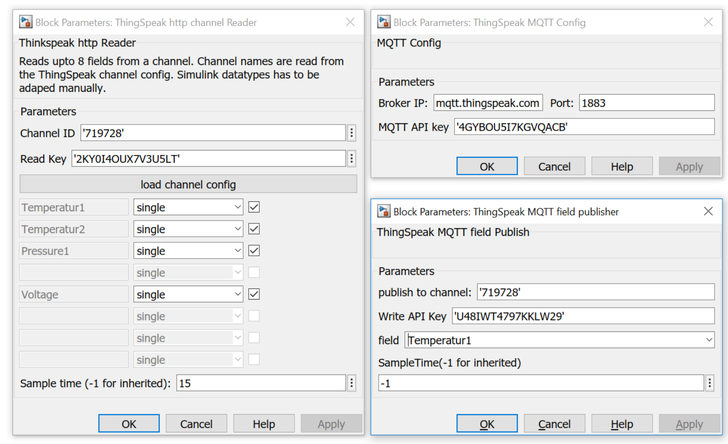


Figure 5: Examples of config masks

Example

Figure 6 shows an example application. It writes 4 sensor values to a ThingSpeak channel in the left half. The http based block to write a complete channel is used. The upper right quarter reads back the same channel from ThingSpeak and displays the values in a graphical LC display. Again, the http based block is used. The lower right part contains initialisation blocks for Ethernet and SPI. Examples using MQTT publish and subscribe blocks are included within Mbed Target which is available at github [4].

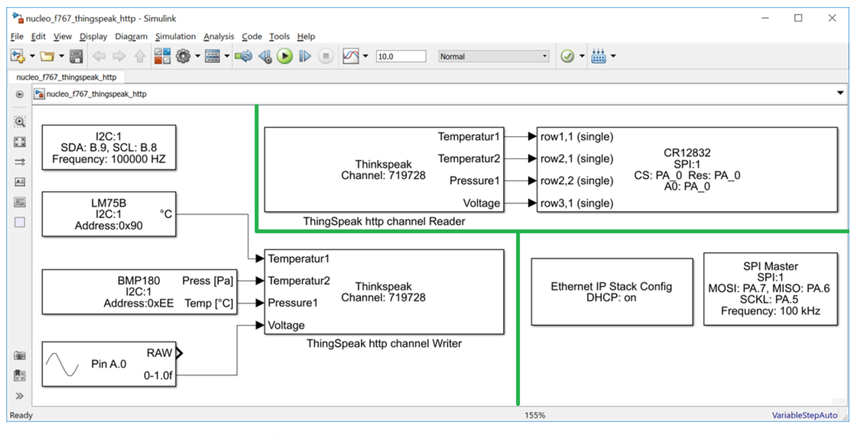


Figure 6: Application example

Conclusion

ThingSpeak is already integrated into MATLAB, for local data handling as well as cloud computing. The proposed extension of Mbed Target, the ThingSpeak block set, completes this at the other end of the communication line, at the IOT node side. Comprehensive prototyping for the complete data flow based on MATLAB/Simulink everywhere is now possible.

[1] https://github.com/ATM-HSW/mbed_target (accessed 01.06.2019)
[2] https://de.mathworks.com/hardware-support/stmicroelectronics.html (accessed 01.06.2019)
[3] https://www.st.com/content/st_com/en/products/development-tools/software-development-tools/stm32-software-development-tools/stm32-utilities/stm32-mat-target.html (accessed 01.06.2019)
[4] https://www.st.com/en/evaluation-tools/stm32-mcu-mpu-eval-tools.html (accessed 01.06.2019)
[5] https://github.com/ATM-HSW/thingspeak4mbed (accessed 01.06.2019)

Indoor Position Data Aggregation and Indicator Extraction for Production and Logistic Issues with MATLAB®

FHWS University of Applied Sciences Würzburg-Schweinfurt | Faculty Business and Engineering | https://fwi.fhws.de
Stefan Kuhn, Gerald Barthelmes, Christoph Bunsen, Jan Schmitt
Institute Digital Engineering (IDEE)
Contact Person | Prof. Dr.-Ing. Jan Schmitt| jan.schmitt@fhws.de | Phone +49 9721 940-8594

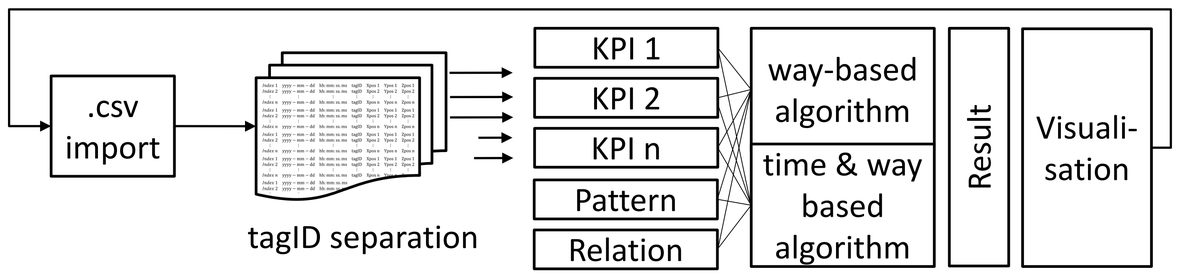
Current Situation

- Location-based services via GPS are widespread in society e.g. smartphone tracking
- In enclosed environments (e.g. assembly hall) indoor positioning systems are available (e.g. WiFi, UWB or Bluetooth), but are not used comprehensively
- Important process information can be derived from position data to extract general key-performance-indicators (KPI) on the shop-floor level of a company
- UWB/WiFi Indoor Positioning System (IPS) consisting of 4 anchors (base points) and a number of tags
- An assembly-line scenario, where the data are aggregated and processed with MATLAB® shows the potentials of the IPS technologies for process management in production/intralogistics

Approach and Systematics

- IPS data can provide much information about objects “moving through” processes
- MATLAB® is utilized for data aggregation in way- and time-based series for each tag separated by the tagID to derive KPIs and pattern about production/logistic processes
- The format for the broadcasted data *D* is

$$D = \begin{pmatrix} Index\ 1 & \text{yyyy} - \text{mm} - \text{dd} & \text{hh:mm:ss.ms} & \text{tagID} & \text{Xpos 1} & \text{Ypos 1} & \text{Zpos 1} \\ Index\ 2 & \text{yyyy} - \text{mm} - \text{dd} & \text{hh:mm:ss.ms} & \text{tagID} & \text{Xpos 2} & \text{Ypos 2} & \text{Zpos 2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ Index\ n & \text{yyyy} - \text{mm} - \text{dd} & \text{hh:mm:ss.ms} & \text{tagID} & \text{Xpos n} & \text{Ypos n} & \text{Zpos n} \end{pmatrix}$$



Objectives

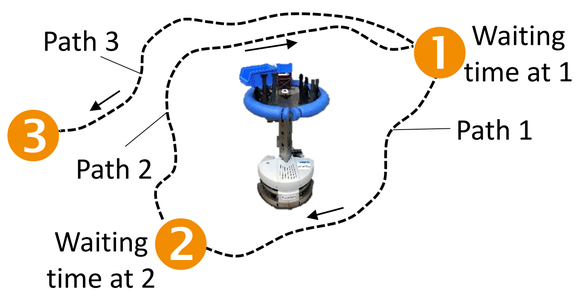
- Use Cases are realised within the c-factory of the FHWS
- c-factory is a concept factory for demonstrating technologies of an Industry 4.0
- Assembly of a customizable toy car with networked production machines

Use Case 1 - Logistics

- The automatic guided vehicle system is equipped with a tag
- The way from ❶ to ❷ and back with different trajectories and to the assembly workplace ❸

Use Case 2 - Assembly

- Assembly of different variants of a toy car by different operators
- Tag is place at the assembly object



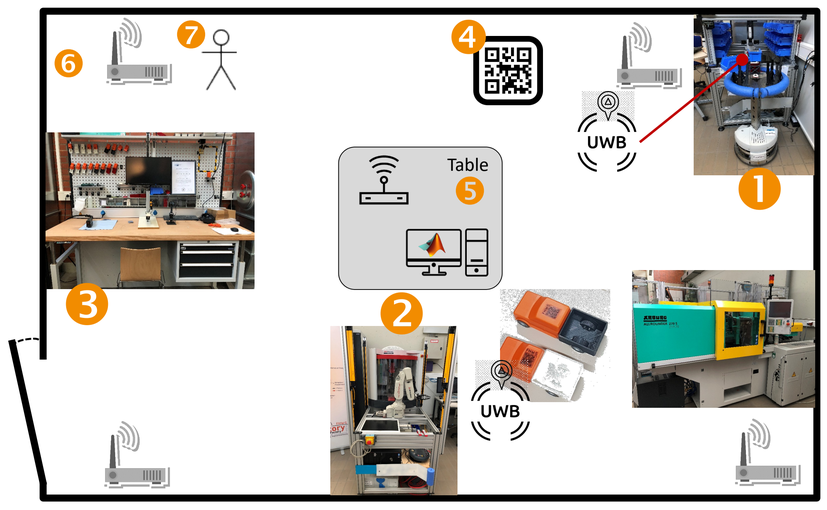
5 → 2 → 1 → 4 → 3 → 6/7

IT-Architecture & Testbed

- Fixed anchors and moving tags
- Tags transfer data via UWB, anchors via WiFi

Anchor
Tag
Router

❶ Stock with automatic guided vehicle system ❷ CNC mill with handling robot ❸ Manual assembly workplace ❹ Label printer ❻/❼ Quality Assurance



Results and Discussion

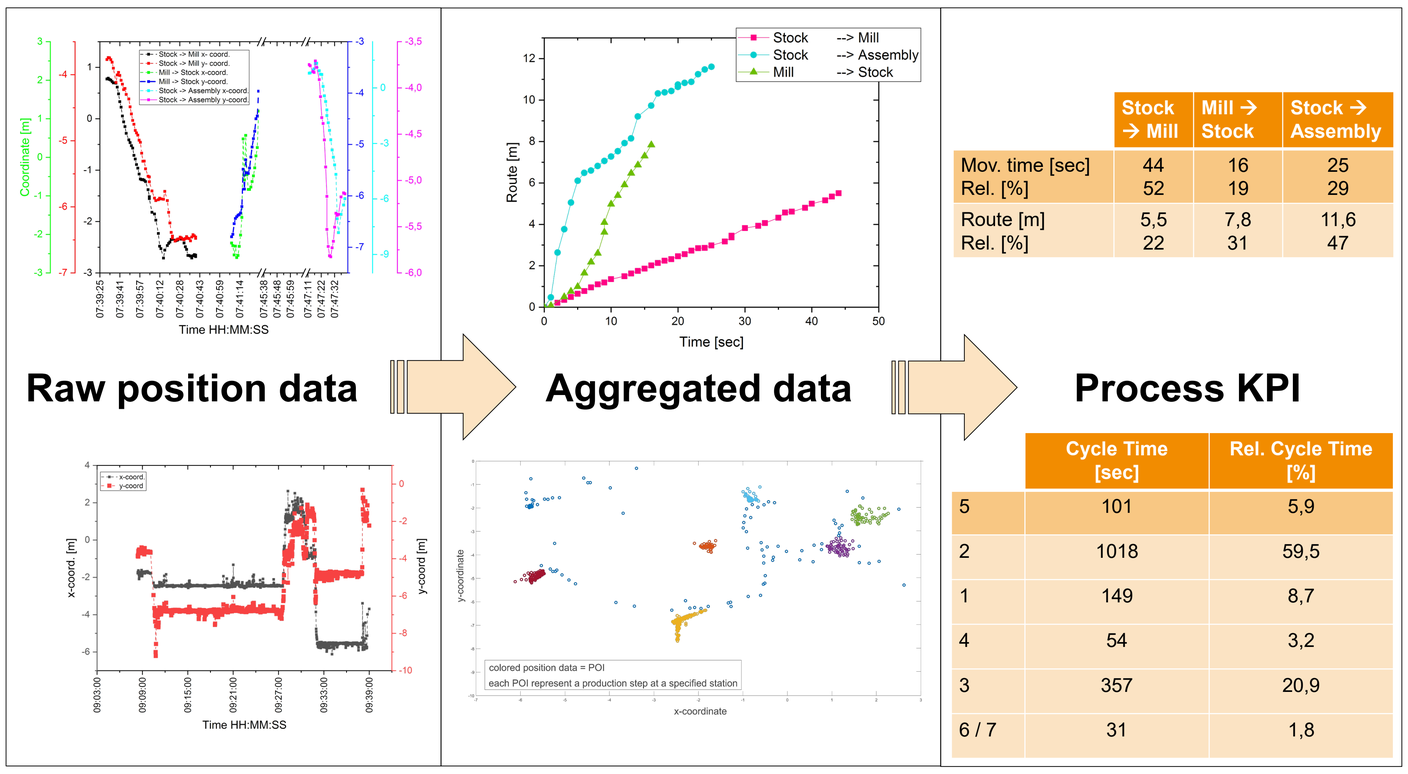
- MATLAB® is used to aggregate & visualize “real time” data
- Import raw-data, algorithms to derive process data/KPIs

Use Case 1 - Logistics

- Raw data show the 3 different routes with x/y-coordinates over the timeline
- Data are aggregated to the route lenght in a time period
- KPIs are the moving time/route length e.g. in relation to the entire way time

Use Case 2 - Production

- Raw data show the time based pattern of assembly in x/y
- The data are aggregate to a 2D-map and the POI are defined and automatically identified and marked
- KPIs indicate abs./rel. cycle time for each production step



	Stock → Mill	Mill → Stock	Stock → Assembly
Mov. time [sec]	44	16	25
Rel. [%]	52	19	29
Route [m]	5,5	7,8	11,6
Rel. [%]	22	31	47

Process KPI

	Cycle Time [sec]	Rel. Cycle Time [%]
5	101	5,9
2	1018	59,5
1	149	8,7
4	54	3,2
3	357	20,9
6 / 7	31	1,8

FH·W·S **University** of Applied Sciences **Würzburg-Schweinfurt**

A Dynamic Sensor Fault Detection and Identification System in IoT using MATLAB and Simulink

Ahlam Mallak, Madjid Fathi
Institut für Wissensbasierte Systeme und Wissensmanagement, Universität Siegen, Germany

Abstract: This work represents a detailed Anomaly Detection (AD) system, that tackles a wide majority of sensor faults. In other words, we are offering a sensor Fault Detection and Identification (FDI) system for IoT. The created machine learning system is evaluated and tested using a dynamic simulation tool/software for case-study generation. Where, a full simulation of a Mobile Wireless Sensor Network (MWSN) of an indoor system is established using Simulink. Then a Graphical User Interface (GUI) is created with MATLAB. In order to, dynamically change the simulated system, generate various scenarios to test and evaluate the AD model created, and provide an open source, customized and a user-specific prototype to any other machine learning task that requires customized WSN datasets.

Introduction

Background

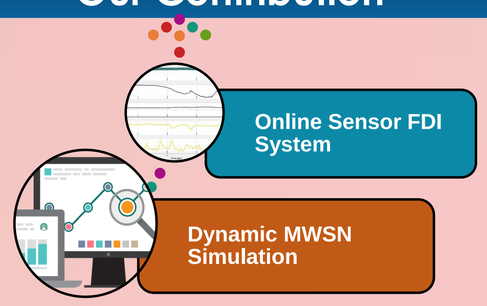
Anomaly Detection (AD) is the process of finding odd, extraordinary or unusual patterns in a given data.

MWSNs can be defined as a Wireless Sensor Network (WSN) in which the sensor nodes are mobile.

Problem Statement

MWSNs Challenges	MWSNs FDI Challenges
MWSNs are so important and needs to be studied.	Most FDIs for MWSN are domain-specific.
MWSNs Have endless possibilities of faults and problems.	Most FDIs for MWSN are done offline.
MWSN needs to be researched with ML, but there is a shortage of use-cases online to use.	Most FDIs for MWSN and needs labels/supervised to perform the diagnosis.
Even if we find good use-cases, it is hard to find ones that are 100% customized to the problem.	Most FDIs for MWSN are so complex and difficult to distribute

Our Contribution



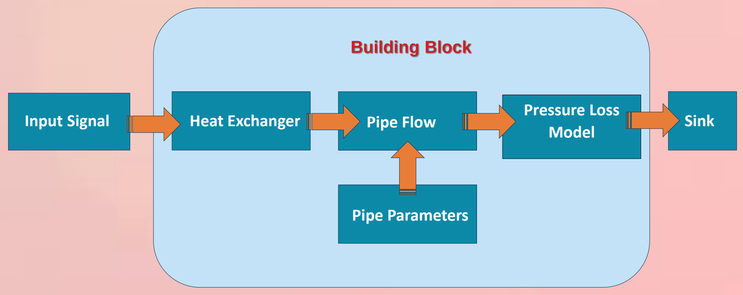
Methods & Experiments

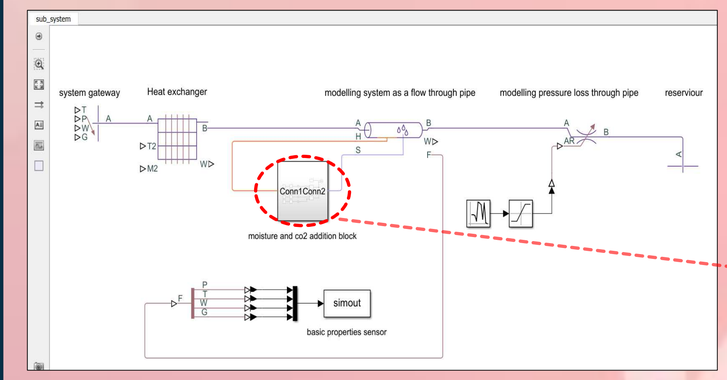
MWSN Model

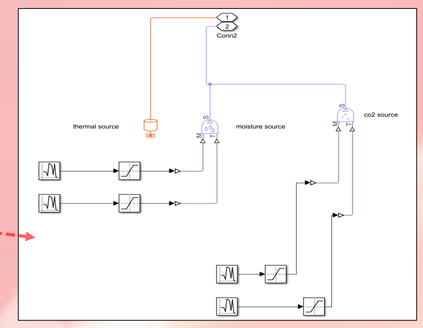
The simulation part represents a dynamic, user-specific simulation tool for case-study generation in wireless sensor networks. This part has two main phases; the first phase is to establish the simulation model for an indoor MWSN using **Simulink MATLAB**. The second phase is to create the user-friendly GUI.

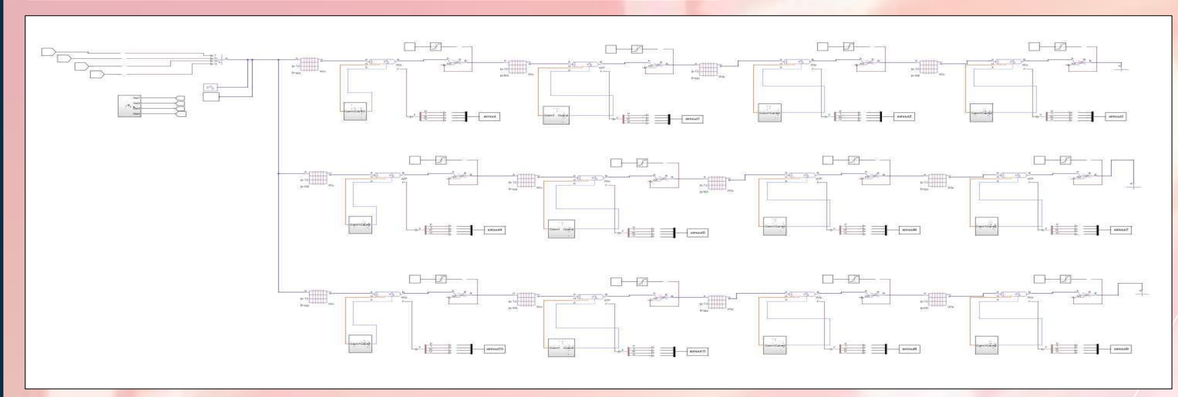
Indoor MWSN System Using Simulink

A moist air model is created replicating real world weather fluctuations using **moist air library**.

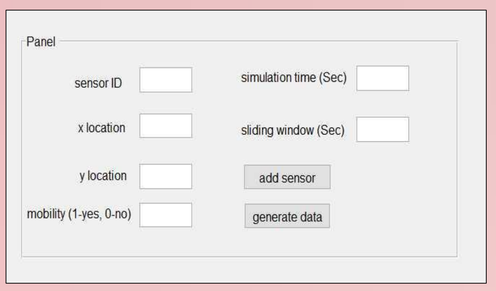


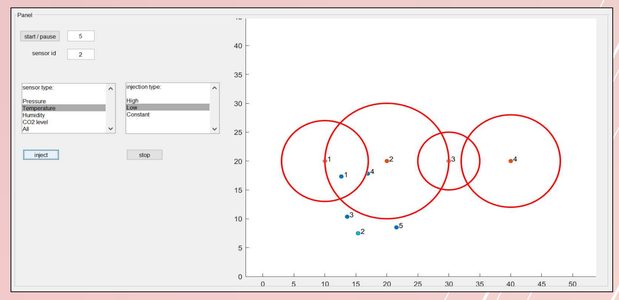






GUI Implementation For More Customized Options

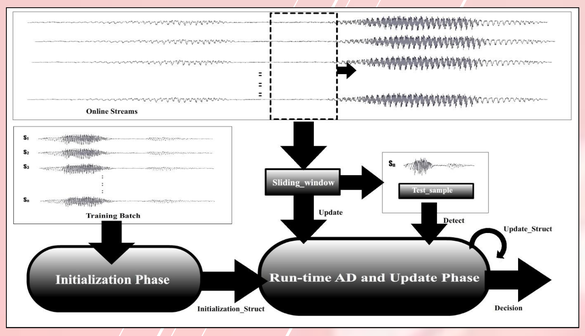




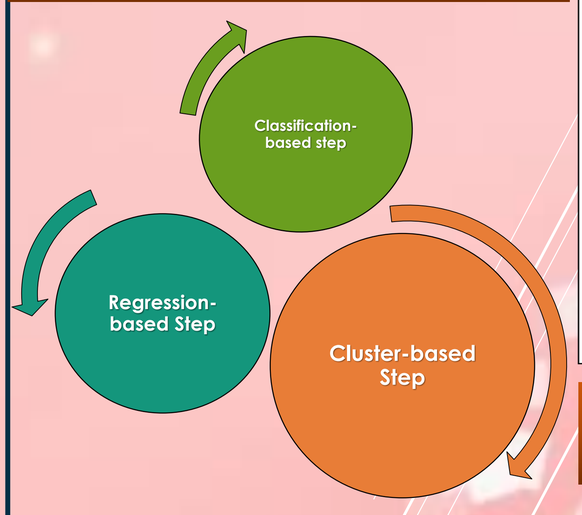
Machine Learning Sensor FDI System Using MathWorks®

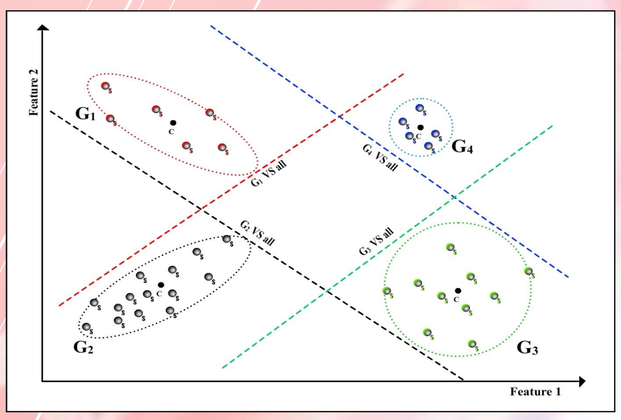
AD System Overview

The main phases of the Sensor FDI framework . 1) **The initialization phase**. 2) **the run-time AD and update model phase**



Initialization Phase/Offline learning





Real-time Phase/Online Detection and Update

Step1:
Find Warning Sensors: Check the trained clusters. If all sensors still belong to the same group OR not.

Step 2:
Check the severity of the warning sensor(s) by reconstructing the best-case/healthy signal of the warning sensor and compare to the current warning signal.

Step 3:
Check the severity of the warning sensor by reconstructing the best case/healthy distance of the warning sensor and the decision boundary of its original cluster. And compare it with the current distance with the decision boundary.

Conclusion

MATLAB has shown high performance of applicability for both online and offline machine learning schemas. Moreover, it was effectively able to model MWSNs using **Simulink.**

More Projects:

The following are some more projects offered by us. Some of them are still running and others are already done. Those Projects use(d) MATLAB for various purposes, partially or entirely.

iDev40	Semi40	Deutsche Edelstahlwerke	Lanxess
Smart process-driven knowledge graph creation using integrative text mining	Yield fluctuations for special high-volume products – intelligent approaches for fluctuation cause analysis	Application of knowledge management methods to optimize heat treatment in the steel industry	Holistic IT management concept for organizational and technical infrastructures
Green KM	**WiRPro**	**Elmos AG**	**Airbus**
Green Knowledge Management zur Unterstützung ökologischer Nachhaltigkeit	Extending Product Lifecycle Management by knowledge-based feedback of product use information into product development	Adaptive Intelligent Analytical Framework for Interrelation Discovery in Production Lines	Multi Domain Strategy for Data Integration of Business Processes as a Controlling Instrument using Data Mining & Visualization

ACKNOWLEDGMENT
This work was supported by the DFG research grants LO748/11-1 and OB384/5-1.
DFG Deutsche Forschungsgemeinschaft

CONTACT
Ahlam Mallak.
Email: Ahlam.Mallak@ymail.com
Mobile: +4915229488004

DETECT LEARN SCHEDULE ADISTES
UNIVERSITÄT SIEGEN
MATLAB® & SIMULINK®
MathWorks®

AI based virtual sensor & durability engineering

load and lifetime monitoring for cars

engineer

-fleet information
-representative profile

development vehicle

measurement

rapid prototyping

speedgoat
real-time simulation and testing

model design

frequency analysis

system identification

AI

sensor fusion algorithm

SIMULINK®

Simulink coder
embedded coder

C code
AUTOSAR &
MISRA compliant

POLYSPACE®
Code Verification Products

code certification

python package export

MATLAB compiler SDK

end user

-damage estimation
-statistics

web interface

cloud application

data analytics
- predictive maintenance
- representative usage

4G

series car fleet

COMPREDICT's virtual sensor and durability engineering platform uses AI to predict failures of automotive components and understand the failure mechanisms. No additional sensors or hadware needed. This is a key differentiation. We analyze and enhance the data, which is available in every series production vehicle. With our data driven approach, we can exactly predict failures of mechanical components and of electric components. Furthermore, our methods enable lightweight design and speed up the development process of automotive OEMs and suppliers.

COMPREDICT

Defining Component & Driver **Analytics** with Software-Based Load and Lifetime Monitoring

Neural Networks

Raban Seitz

What is Machine Learning?

- Machine learning is a technique to teach computers to learn from experience. Machine learning algorithms use computational methods to built models without relying on a predetermined equation as a model. Therefore, Machine learning algorithms find natural patterns in data, make predictions and iteratively change parameters in order to create a well-suiting, but general mathematical model.

Categories

- <u>Supervised Learning:</u>
  Supervised Learning algorithms create a function that maps an input to an output based on example input-output pairs.
- <u>Unsupervised Learning:</u>
  Unsupervised Learning algorithms find previously unknown patterns in data sets without labels
- <u>Reinforcement Learning:</u>
  Reinforcement learning algorithms try actions in an environment to maximize the reward. The focus is on finding a balance between exploration of “unknown territory” and exploitation of current knowledge.

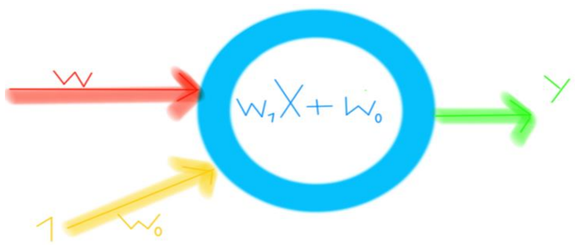
=> Neural Networks can be used for all three categories of Machine Learning.

Neural Networks

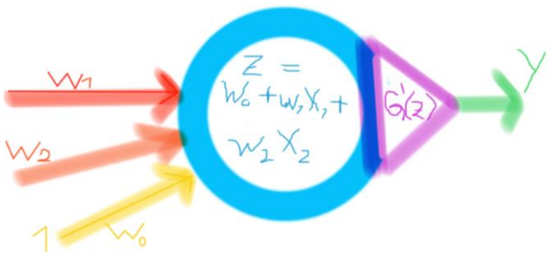
Neural Networks:

Neural networks are collections of neurons, which model the neurons in a human brain. Each connection, like synapses in a human brain, can transmit a “signal” from one neuron to another. An artificial neuron that receives a signal can process it and pass the signal to another neuron connected to it.

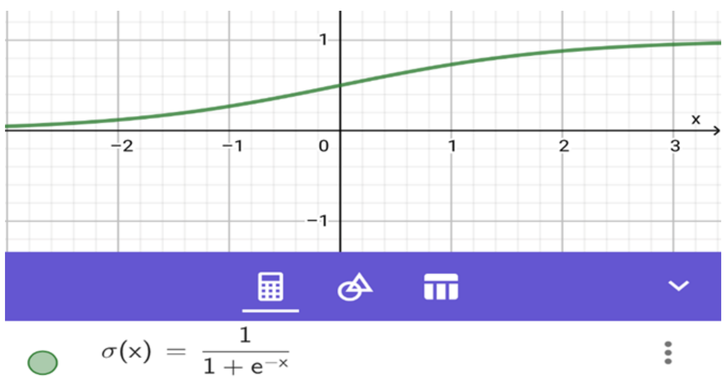
- 1. Linear neuron:



- This linear neuron (above) has one input feature (x), two weights (w1, w0) and one output (y). The output y is a hypothesis created by the multiplication x*w1 plus w0. y=x*w1+w0
- 2. Logistic neuron:

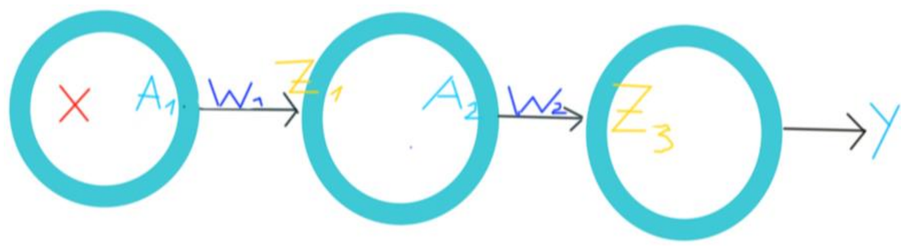


- This is a logistic neuron. The first step is similar to the linear neuron: Z=x1*w1+x2*w2+w0, but in the next step we use an activation function.
- In this case we use the sigmoid function:



- The reason to use the sigmoid function is that the sigmoid function limits the output of this neuron to a number between zero and one and because of this, outlier, training datapoints that are far away from the rest, do not change our parameters that much, as they would do in a linear neuron.

Forward Pass



Above you can see the three layers again, but just one path and how the input data is passed on:

$A_1 = X$

$Z_1 = W_{1,2} * A_1$

$A_2 = \text{sigmoid}(Z_1)$

$Z_2 = W_{2,3} * A_2$

$A_3 = \text{sigmoid}(Z_3)$

$Y = A_3$

Simplified Cost Function

$$\mathbb{C}_{(w_0,w_1)} = \frac{1}{2 \cdot n_{samples}} \sum_{n=1}^{n_{samples}} \left(\hat{y}(x_n) - h_{(w_0,w_1)}(x_n)\right)^2$$

- This function creates the squared error (difference between prediction and real value) for every sample n ∈ nsamples.
- Then it creates the sum of all errors and divides them by the amount of samples to get the middle error for each parameter.

Error Gradient

- In order to get the error gradient, we do the partial derivation in direction w1:

$$\frac{\partial \mathbb{C}}{\partial w_1} = \frac{1}{n_{samples}} \sum_{n=1}^{n_{samples}} \left(h_{(w_0,w_1)}(x_n) - \hat{y}(x_n)\right) \cdot x_n$$

- and after that in direction w0:

$$\frac{\partial \mathbb{C}}{\partial w_0} = \frac{1}{n_{samples}} \sum_{n=1}^{n_{samples}} \left(h_{(w_0,w_1)}(x_n) - \hat{y}(x_n)\right)$$

- The error gradient gives us the direction for changing w1 and w0.
- The learning rate LR is the step-size:

$$w_0 := w_0 - LR \cdot \frac{\partial \mathbb{C}}{\partial w_0}$$

$$w_1 := w_1 - LR \cdot \frac{\partial \mathbb{C}}{\partial w_1}$$

Rupprecht-Gymnasium München
Naturwissenschaftlichtechnologisches
und Sprachliches Gymnasium

ZUSAMMEN.WACHSEN.

Facial Recognition with Convolutional Neural Networks

Kai Angnis

My W-Seminar topic

In my W-Seminar, I will combine a convolutional neural network, that is good at recognizing patterns in different regions of an image, with a siamese network, a machine learning structure, that can take two inputs and returns a similarity value for them.
This neural network is going to be trained on the recognition of faces, in order to distinguish a pair of images that shows the same person from a pair of images, that shows different people. A neural network like this could be used as a way of access control.

Neural Network:

- Machine Learning approach
- Several layers, like in this case convolutional layers and siamese layers, are used to classify inputs, that don't have to be linear separable
- Extracts features from the input without the help of a human
- Can be modeled as a parameterized function with the weights as parameters
- Weights are often trained via backpropagation [1]

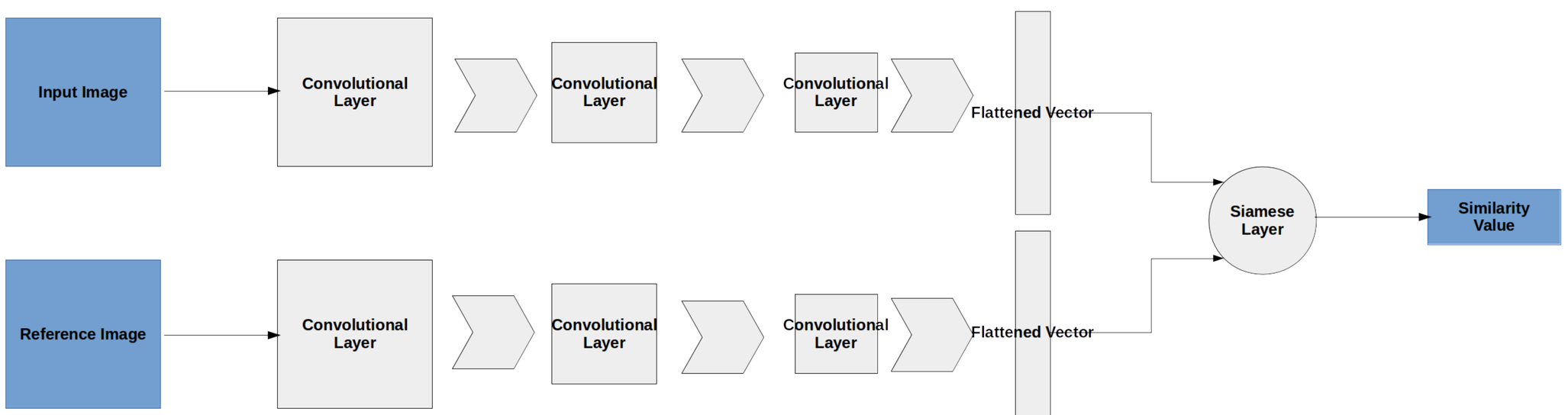


Fig. 1: The planned architecture of my neural network

Convolutional Layer

$$o_i = f_{activ}\left(\sum_j o_{i-1} * k_{ij} + b_i\right)$$

- Uses convolution in order to "slide" a filter over the image or the previous layer
- More than one kernel is used in order to detect more than one feature in one layer
- Detects occurrences of the filtered objects anywhere on the image [2]
- Usually ReLU is used as the activation function:

$$f(x) = max(0, x)$$

- With the activation function, the trained classes can be separated in a nonlinear way [3]

Siamese Layer

$$o = ||o_1 - o_2||$$

- The Inputs of the siamese layer are the two Images that where fed through the convolutional layers with the same weights
- The L1 distance can be used to return a guess on the similarity of the two inputs, the resulting vector is passed through a fully connected layer to obtain a single value as the output
- Uses a sigmoid function as the activation function in order to obtain a propability value between 0 and 1[4]:

$$f(x) = \frac{1}{1 + e^{-x}}$$

A Convolutional Layer in MATLAB

```
function features = convlayer(images, W , b)
[~,~,imagecount]=size(images);
filtercount = size(b);
for image = 1:imagecount
  for filter = 1:filtercount
      convimg = conv2(images(:, :,image),rot90(W(:,:,filter),2),
'valid');
      convimg = convimg + b(filter);
      features(:, :, filter, image) = convimg;
  end
end
features=max(0,features)
end
```

Training the Network

- In order to be able to interpret the output of the neural network as a probability, the error function, in this case the cross-entropy function is partially derived in respect to the different weights. [5]
- The weights of a given layer are updated by applying the corresponding derivative on a datatapoint of the training data. This value is then multiplied by the learning rate, in order to avoid overshooting an error minimum

Sources:
1.LeCun, Y. and Bengio, Y., 1995. Pattern recognition and neural networks. The Handbook of Brain Theory and Neural Networks. MIT Press, Cambridge, MA.
2. Y.LeCun Yoshua Bengio, 1998 Convolutional Networks for Images, Speech, and TimeSeries
3.Zhang, K., Zuo, W., Chen, Y., Meng, D. and Zhang, L., 2017. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. IEEE Transactions on Image Processing, 26(7), pp.3142-3155.
4. Koch, G., Zemel, R. and Salakhutdinov, R., 2015, July. Siamese neural networks for one-shot image recognition. In ICML deep learning workshop (Vol. 2).
5. Gundogdu, Batuhan,2017 Keyword Search for Low Resource Languages.

ZUSAMMEN.WACHSEN.
Rupprecht-Gymnasium München
Albrechtstraße 7
80636 München
mint@rupprecht-gymnasium.de

MathWorks
Adalperostr. 45, 85737 Ismaning
www.mathworks.de

MINT-EC
Poststraße 4/5, 10178 Berlin
info@mint-ec.de

MATLAB EXPO 2019

Hochschule Esslingen
University of Applied Sciences

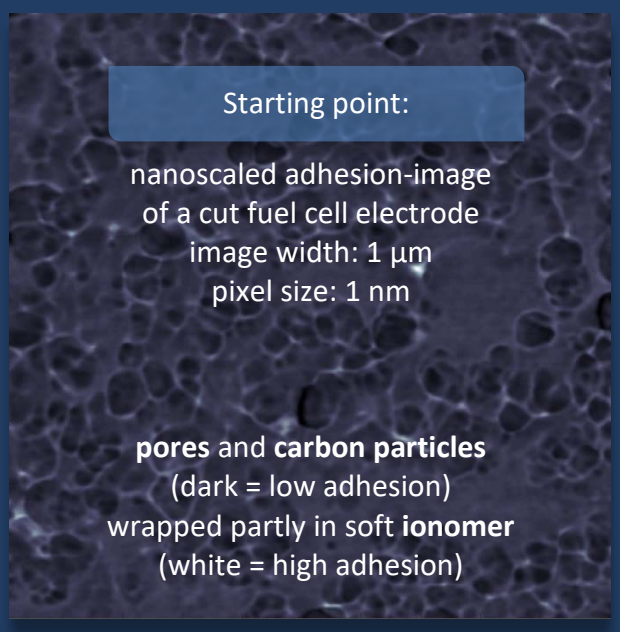
Image Processing in Electrochemistry

Measuring ionomer layers in a fuel cell electrode

Jan-Frederik Heger, Prof. Dr. Andreas Helfrich-Schkarbanenko, Fakultät Grundlagen, HS Esslingen

MATLAB EXPO, 02. Juli 2019, MÜNCHEN

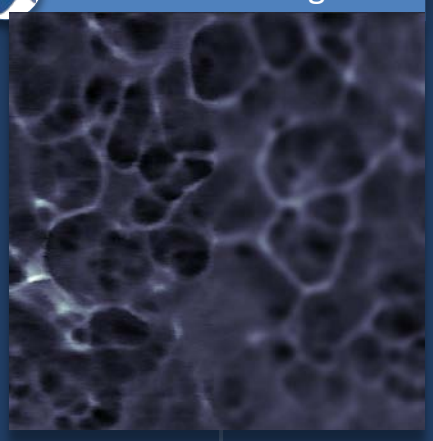
The purpose of an atomic force microscope (AFM) is to analyse material by its physical, chemical and electrical properties on a nanometer scale. Thus a tip, attached on an oscillating cantilever, is being moved over a sample. On the cantilever a laser is reflected towards a split photodiode. By these measurements the influence of the sample on the tip can be detected. Different material information can be calculated in this way. Examples for these gathered information are topography and adhesion – which are used in this presentation to detect the ionomer inside a fuel cell electrode and calculate its layer thickness.



A proton-exchange membrane fuel-cell persists of a membrane surrounded by two electrodes, which are constituted from platinum particles, sticking on carbon particles and ionomer which is partially in between the particles. Of a big importance is the ability of the electrode material to transport both the gases, the ions and the electricity.

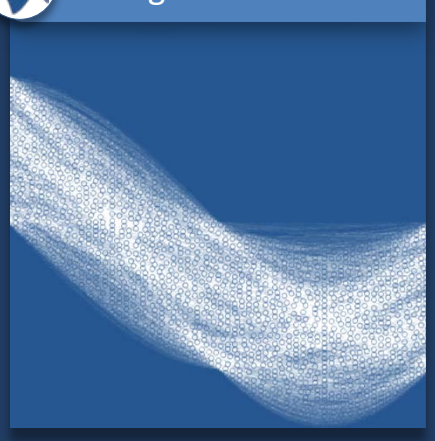
A still unanswered question is the distribution of the ionomer and especially its layer thickness. To understand the chemistry and physics happening inside a fuel cell on a nanometer scale this question needs to be answered.

Gauß-Filtering



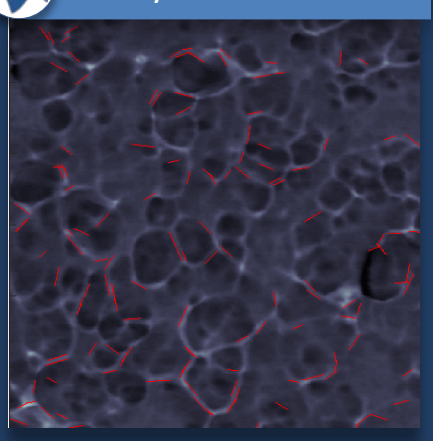
The first stage of the algorithm ist the preprocessing of this image – it is **gauß-filtered** and ist edges are detected (canny algorithm).

Hough Transform



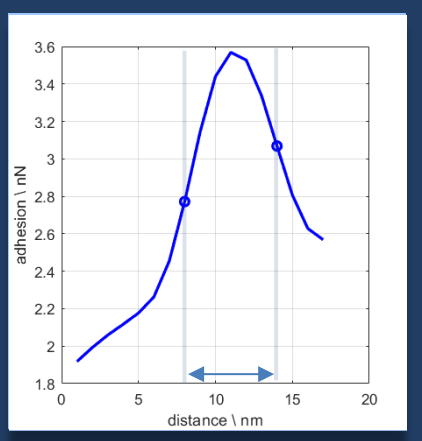
In the second stage the locations of the ionomer layers are detected: with a **Hough Transform** the line segments are identified. Those segments are **filtered** by its geometry.

Layer Position



Now the layer position is determined. In a third step for every segment a profile is calculated orthogonal to the segment in a loop.

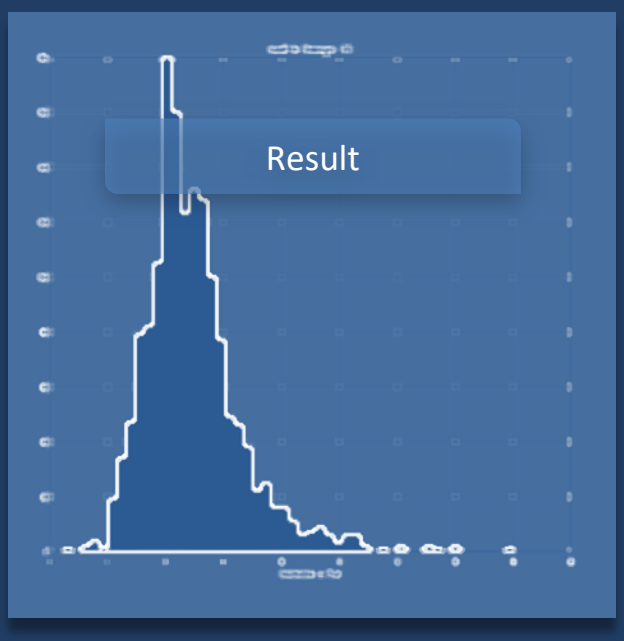
Layer Profile Analysis



The profile is now smoothed and numerically derived. If two **inflection points** can be found in this profile, the corresponding **ionomer layer** can be calculated as the distance between those two points.

After all the detected segments are evaluated, the distribution can be analysed and compared with results from other images. A typical histogram is shown in the left.
Here the average ionomer thickness is around 6 nm.

hs-esslingen.de/~jfheger/MATLAB-EXPO-2019.pdf



Next modifications of the algorithm

- Automatic determination of the parameters for the Hough Transform
- Batch processing applied on several images
- Filter for horizontal distortion in the AFM images due to shifting tip
- Involving the 3D structure of the particles into the layer thickness computation
- Measurements of the particles and pores

HTTP://WWW.HS-ESSLINGEN.DE/HOCHSCHULE/FAKULTAETEN/GRUNDLAGEN.HTML
Hochschule Esslingen // Kanalstraße 33 // 73728 Esslingen // Tel +49(0)711.397-3443 // jfheger@hs-esslingen.de

University of Basel

Dynamo

A generic Matlab-based scientific library
applied to cryo electron tomography

Daniel Castaño-Díez

BIOZENTRUM
The Center for Molecular Life Sciences

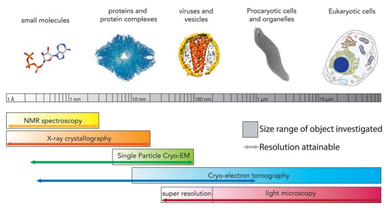
C-CINA

BioEM LAB

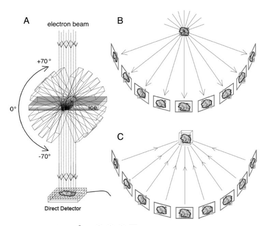
Dynamo

Cryo Electron Tomography

- Visualization of cellular entities
- 3D model integrates a series of tilted views



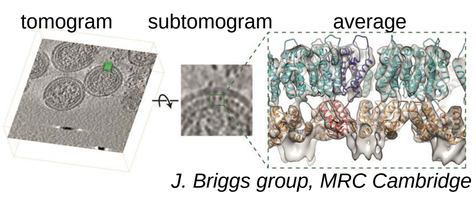
Hutchings et al, 2018



Steven et al, 2005

Subtomogram Averaging

- Enhances the signal of multiple copies of a protein by mutual alignment and averaging
- Allows atomic resolution of *in-situ* structures

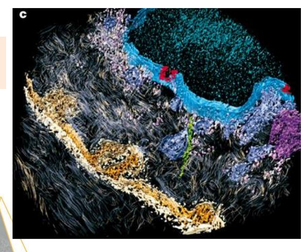


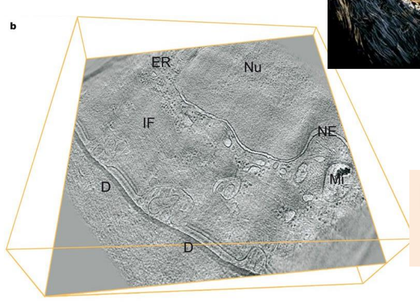
J. Briggs group, MRC Cambridge

The *Dynamo* package

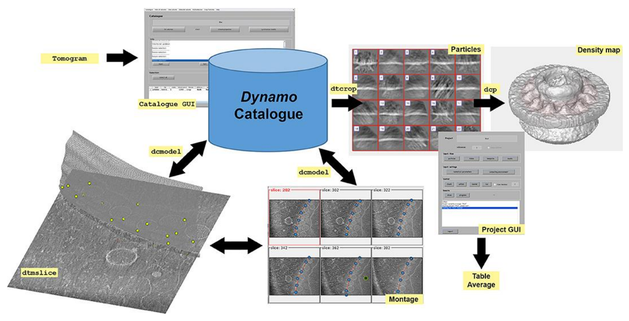
- Covers all aspects of computational tomography
- 800K lines of Matlab,C++ and CUDA code
- GPU and cluster versions
- Hundreds of users worldwide
- Delivered as Matlab package or standalone
- Bianual release for Linux, Mac and Windows
- Several workshops organized during the year

Segmentation

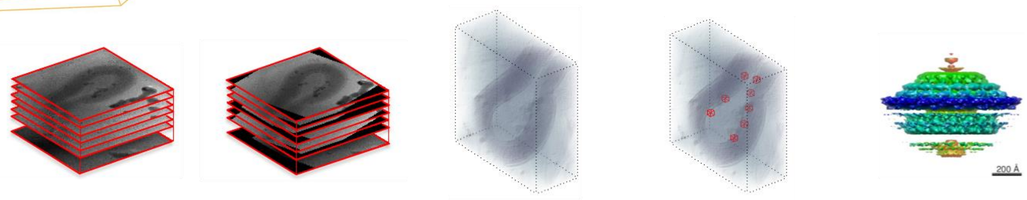




Large 3d data visualization

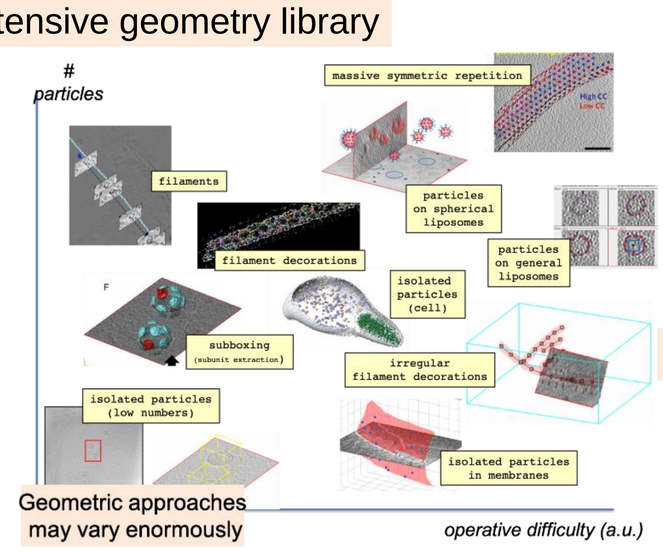


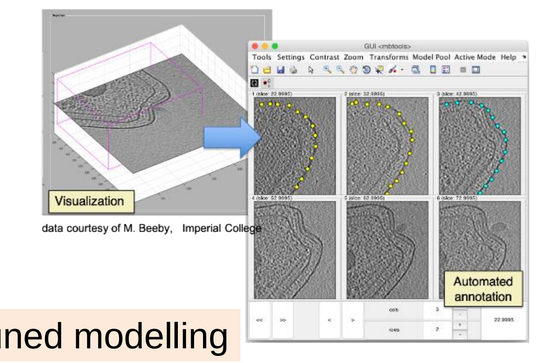
Database for integrated pipelines



Automated shape detection

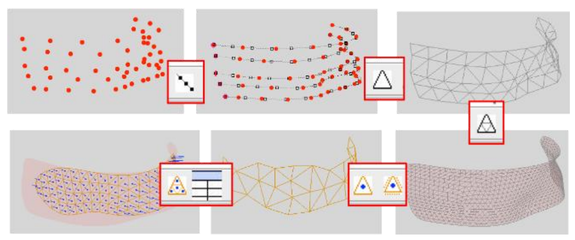
Extensive geometry library





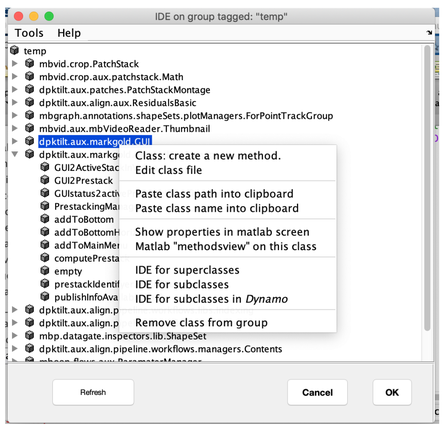
data courtesy of M. Beeby, Imperial College

Fine tuned modelling



MBTOOLS

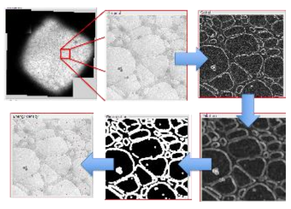
- Generic scientific library running behind *Dynamo*
- **Not** specific for microscopy
- Main functionalities:
  - Customizable 3D visualization
  - Customizable stream visualization
  - Creation of workflows and workflow GUIS
- Advanced programming tools
  - IDE for class oriented development
  - Refactoring utilities



Application example

Workflow for semiautomated detection of sample in electron microscopy grids

Maths are easy: set of basic image processing operations



Problem:

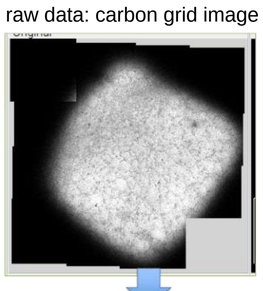
Many different prototypes need to be delivered to end users

End users (microscope operators)
- have no mathematical/computing formation
- need intuitive interaction for visualization and tuning

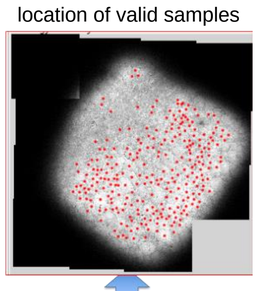
Optimize development time

Approach:

raw data: carbon grid image

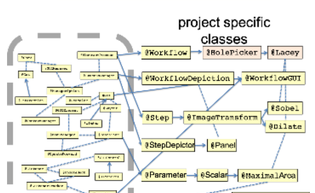


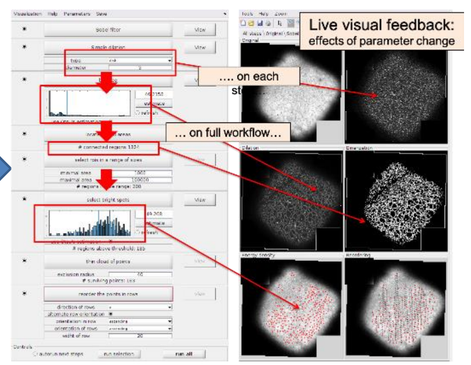
location of valid samples



```
is = mbimg.filters.sobel(I,[3,3]);
id = mbimg.morph.dilate(is,[4,4]);
ib = mbimg.bw.binarize(id,12.56);
cw = mbimg.connectity(imcomplement(ib));
cs = mbimg.connectivity.grepSize(cw,500);
ci = mbimg.connectivity.grepIntensityDensity(cs,140);
xy = mbimg.connectivity.getCenters(ci);
```

MBTOOLS:
class oriented library that maps scripts into a workflow





- GUI produced automatically
- full interactivity for end users
- minimal development effort

Want to work with us?

- Post doc positions available in 2020
- Advanced Matlab skills required
- Contact: daniel.castano@unibas.ch

Want to know more?

www.dynamo-em.org



training workshops organized around the year

Hochschule Konstanz
Fakultät Maschinenbau
Betreuer: Prof. Dr. Jens Weber

Projektarbeit 2

High Speed Thermography Mapping

Leutrim Berisha

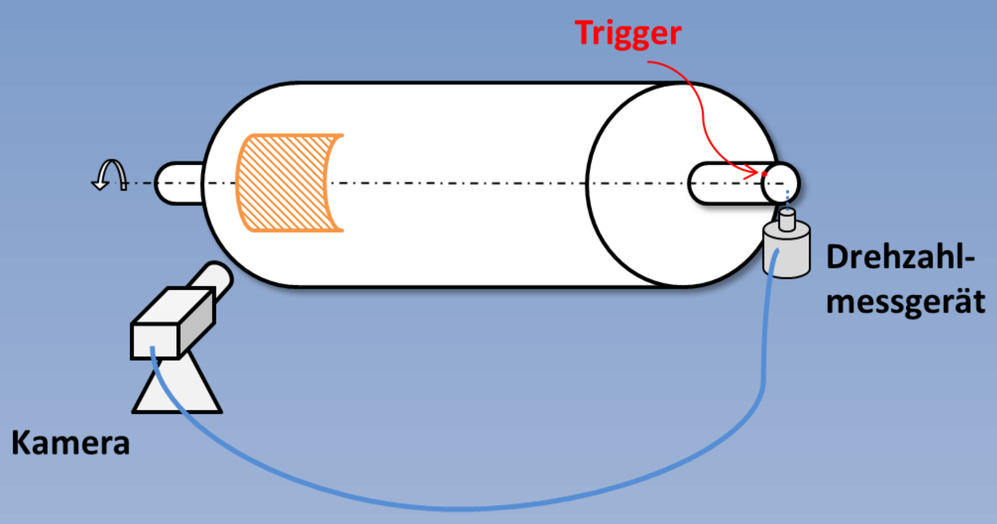
Betreuer: Stefan Natterer
Manager Measurement and Diagnostic Service

Einleitung

Beim High Speed Thermography Mapping (HST Mapping) wird die Oberflächentemperatur eines rotierenden Heizzylinders (dampfbeheizt) mit einer Hochgeschwindigkeitswärmebildkamera erfasst und anschließend innerhalb einer speziell dafür erstellten Benutzeroberfläche (GUI) analysiert. Ziel ist es die abgewickelte Oberfläche des Zylinders auf einer Wärmebildkarte darzustellen. Dieses Messverfahren kommt vor allem bei der Überprüfung eines Yankees zum Einsatz. Beim Yankee handelt es sich um eine spezielle Ausgabe eines Heizzylinders. Er kommt bei der Herstellung von Tissue-Papier zum Einsatz und besitz im Vergleich zu normalen Heizzylindern eine enorme Größe, was eine sehr komplexe Kondensatabfuhr bedingt. Mit Hilfe des HST-Mappings können mögliche Probleme bei der Kondensatabfuhr des Yankee-Zylinders lokalisiert werden. Die Benutzeroberfläche ermöglicht es dem Anwender die von der Wärmebildkamera gelieferten Bilddateien genau zu betrachten und zu bearbeiten. Um ein erfolgreiches Mapping zu gewährleisten, werden Inputinformationen wie der Abstand von der Kamera zur Messoberfläche, der Blickwinkel des Objektivs und die Rotationsrichtung des Zylinders benötigt. Während des Mappings werden bewusst Bilddateien überlagert, um das Überlagerte Bild entsprechend oft zu mitteln. So entsteht eine Wärmebildkarte der rotierenden Oberfläche eines Heizzylinders im stationären Betrieb.

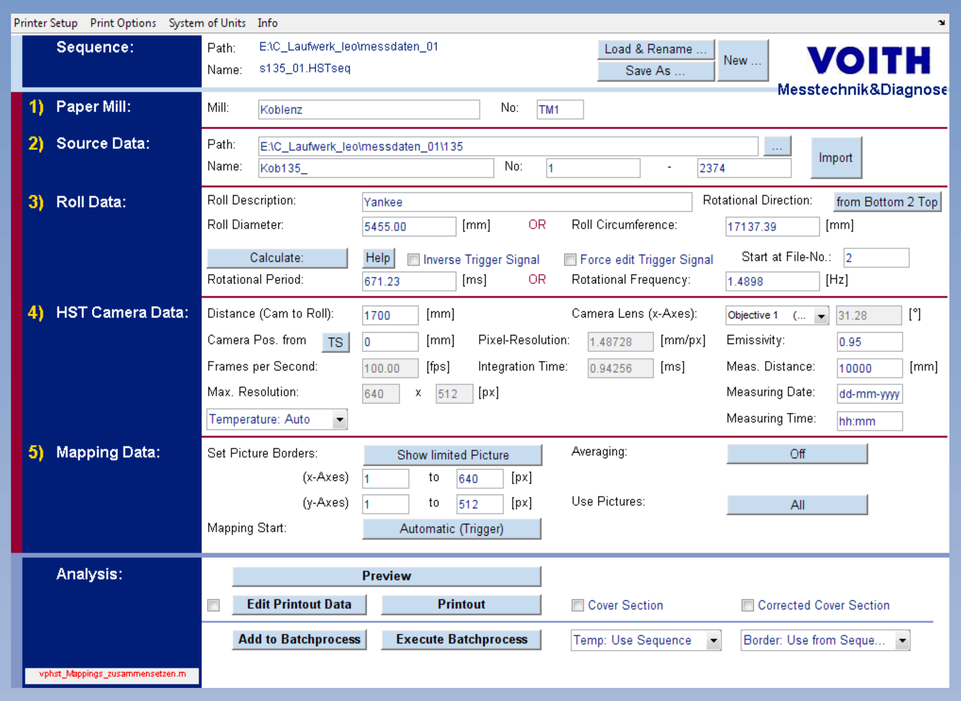
Vorgehensweise

Bildaufnahme



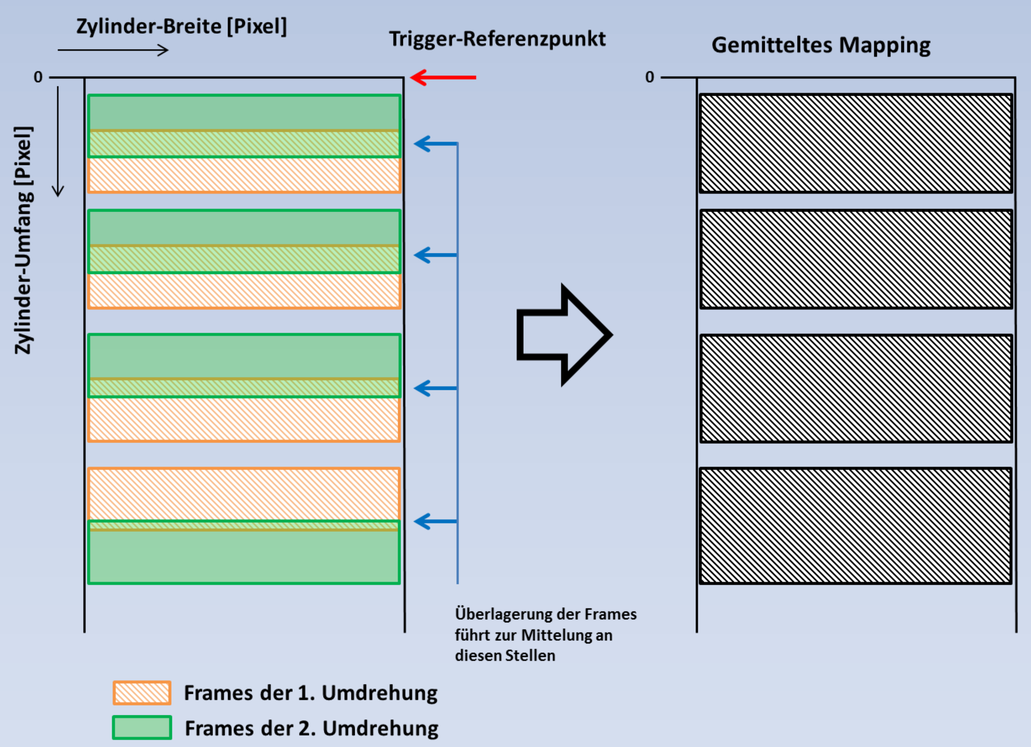
Während der HST-Aufnahmen wird die Drehzahl mit einem Drehzahlmessgerät anhand einer Triggermarke ermittelt. Über einen Impulsausgang werden die Triggersignale gleichzeitig mit der Kamera auf jedem Bild festgehalten. Somit lässt sich im Video genau erkennen, wann der Trigger am Drehzahlmessgerät vorbei zieht. Dieses Triggersignal wird bei der späteren Weiterverarbeitung in Matlab als TTL-Impuls verwendet.

Daten-Import und Bearbeitung



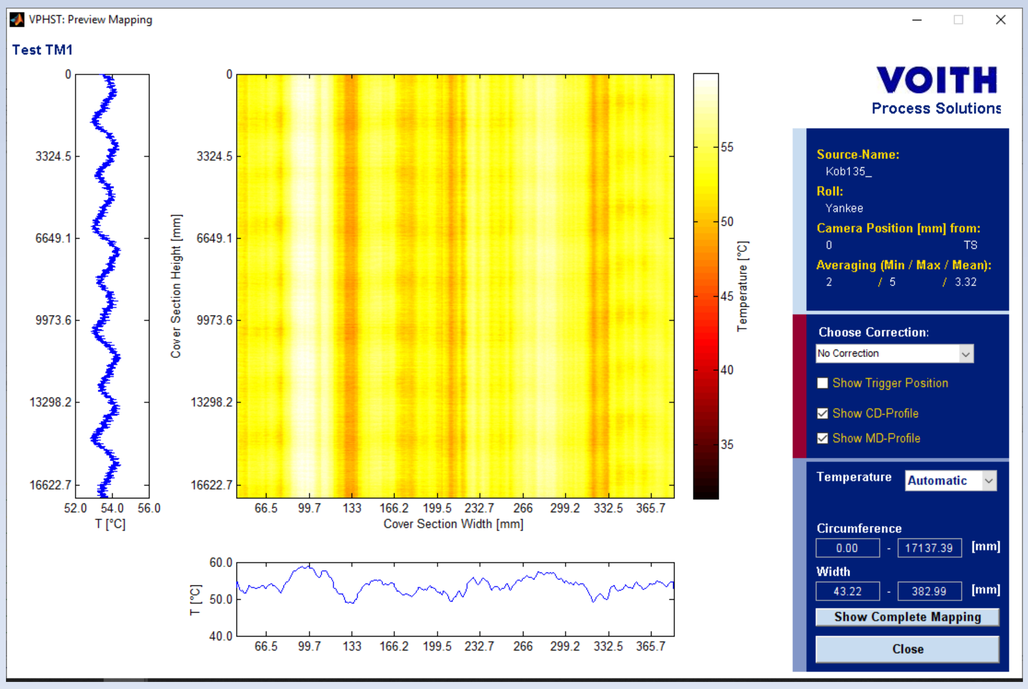
Innerhalb der HST-Benutzeroberfläche werden die Bilddateien importiert und analysiert. Dabei werden aus den Bilddateien wichtige Kamerainformationen wie zum Beispiel die verwendete Framerate, Bildauflösung und Belichtungszeit entnommen. Außerdem wird mithilfe der Trigger-Informationen aus den Bilddateien die Drehzahl des Zylinders bestimmt, um beim Mapping die genutzten Bilddateien aus verschieden Umdrehungszyklen an die richtigen Stellen der gemappten Oberfläche zu setzten. Das Mapping setzt die Zylinderoberfläche anhand der Bilddateien in Form einer Wärmebildkarte zusammen. In der Benutzeroberfläche grenzt der User die relevanten Bereiche der Bilddateien spezifisch ein, um ein korrektes Mapping durchzuführen. Relevante Bereiche der Bilddateien sind Bereiche die aufgrund der gekrümmten Fläche räumlich nicht zu weit Hinten liegen. Somit sind schmale Streifen die mittig auf der Zylinderoberfläche liegen für das Mapping relevant.

Mapping



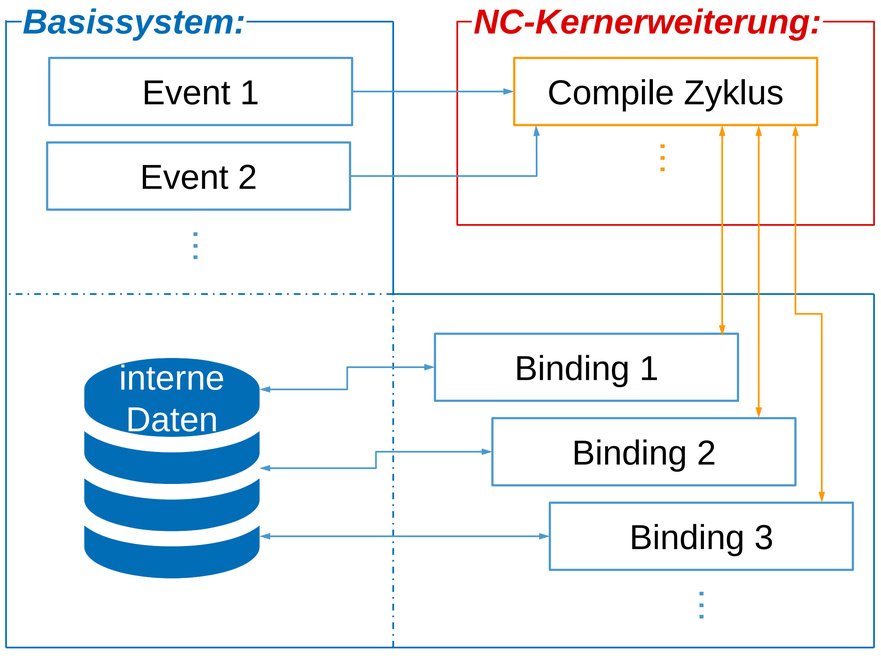
Beim Mapping-Prozess werden fest definierte Bereiche aus allen Bilddateien nebeneinander gesetzt, um so die abgewickelte Oberfläche des Zylinders auf einer Wärmebildkarte darzustellen. Um die Position der schmalen Bildstreifen auf der Zylinderoberfläche bestimmen zu können, sind die Kenntnisse von Drehrichtung und Umdrehungszahl des Zylinders unverzichtbar, wobei die Drehrichtung direkt im HST-Hauptprogramm festgelegt wird. Alle Frame-Aufnahmen wurden zu verschiedenen Zeitpunkten erstellt und weisen somit verschiedene Positionen auf der Zylinderoberfläche auf. An den Stellen, an denen sich mehre Aufnahmen überlagern, wird entsprechend oft gemittelt. Je nachdem wie viele Mittelungen in der HST-Benutzeroberfläche ausgewählt wurden, werden demensprechend viele Umdrehungszyklen zum Mappen genutzt

Ergebnis

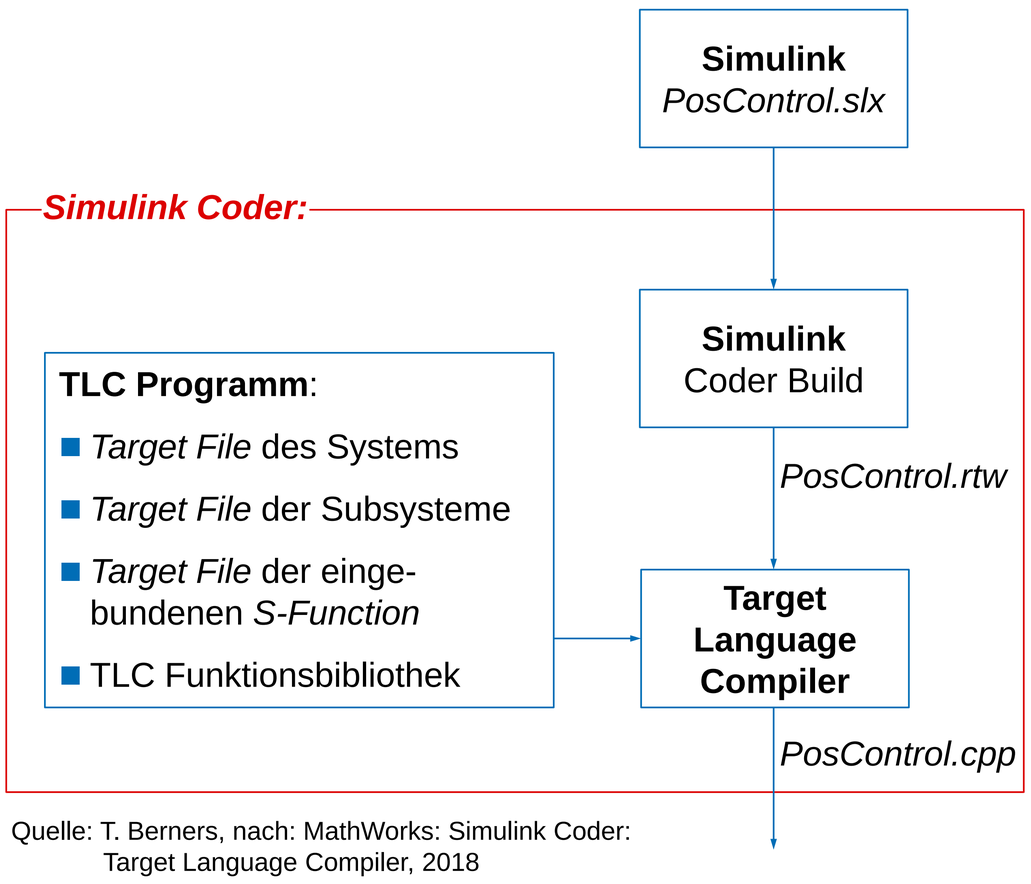


Nach dem Mapping-Prozess wird die gemappte Wärmebildbarte in der Preview-Benutzeroberfläche dargestellt. Hier werden außerdem das Quer- und Umfangstemperaturprofil angezeigt. Mit den seitlichen Optionen ist es möglich, Korrekturen in Quer- und oder Umfangsrichtung anzuwenden. In Kombination mit der Begrenzung des gewünschten Temperatur- und Kartenbereiches lassen sich Probleme der Kondensatabfuhr erkennen. Bereiche in denen die Temperatur ungewöhnlich niedrig ist, weisen auf Ansammlungen von überschüssigem Kondensat hin, welches nicht ordnungsgemäß abgeführt wird.

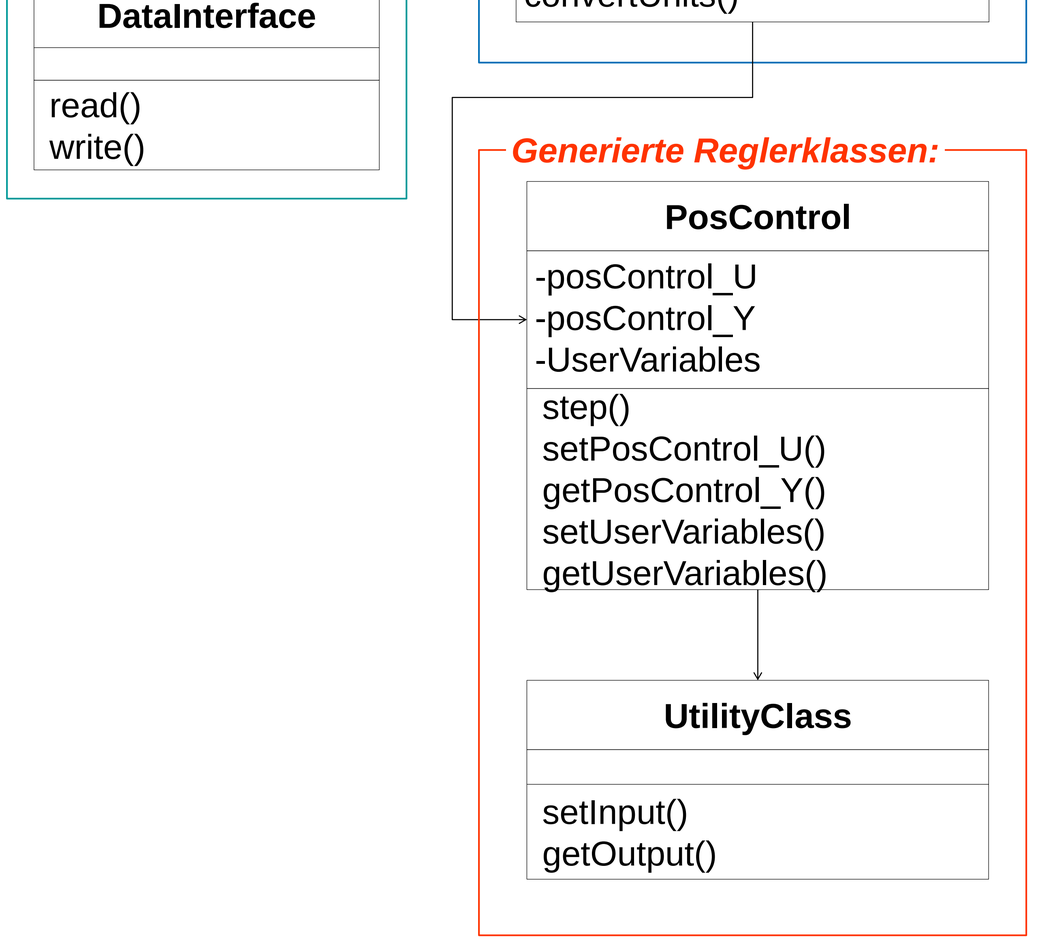
Plattformunabhängige NC-Kernerweiterung mit MATLAB\Simulink



Quelle: Brecher et. al. in: wt Werkstattstechnik online, Vol. 106, 2016, S. 320–324.



Quelle: T. Berners, nach: MathWorks: Simulink Coder: Target Language Compiler, 2018



Quelle: T. Berners, Dissertation (eingereicht, bisher nicht veröffentlicht), WZL RWTH Aachen

Motivation

Die Genauigkeit einer Werkzeugmaschine wird entscheidend von der Lageregelung ihrer Vorschubachsen beeinflusst. Aufgrund der stetig steigenden Leistungsfähigkeit digitaler Steuerungskomponenten bietet sich die Möglichkeit der Integration moderner Regelungskonzepte in bestehende Steuerungsarchitekturen. MATLAB\Simulink liefert im Bereich der Reglerentwicklung viele Unterstützungsfunktionen, die die Auslegung erheblich beschleunigen. Das Problem besteht darin, den in MATLAB ausgelegten Regler in die Lageregleung der numerischen Steuerung (engl.: NC = Numerical Control) zu integrieren. Einige Steuerungshersteller bieten Programmierschnittstellen an, die den Zugriff und die Manipulation von steuerungsinternen Signalen und Daten in Echtzeit ermöglichen. Die Erstellung solcher NC-Kernerweiterung erfolgt jedoch meist durch manuelles Programmieren in einer Hochsprache und ist zeitintensiv, fehleranfällig und erfordert weitreichende Programmierkenntnisse.

Zielstellung

Das Ziel der vorgestellten Toolbox SinumerikRCP (Sinumerik Rapid Control Prototyping) ist es, diesen Programmierschritt zu ersetzen und eine automatisierte Generierung einer lauffähigen NC-Kernerweiterung aus einem Simulinkmodell zu ermöglichen. Dadurch wird ein langwieriger sowie fehleranfälliger Programmierprozess vermieden und gleichbleibende Programmierstandards eingehalten.

Ergebnisse und Vorgehen

Über eine grafische Benutzeroberfläche können die umzuwandelnden Simulink-Subsysteme ausgewählt und projektspezifische Einstellungen vorgenommen werden. Bei der Umwandlung in eine NC-Kernerweiterung verwendet SinumerikRCP den Simulink Coder zur Erstellung des C++ Modellcodes und integriert diesen vollautomatisch in einen steuerungsspezifischen Rahmencode. Anschließend wird der Kompiliervorgang des Gesamtcodes direkt aus MATLAB heraus gestartet.

Um die Lösung nicht auf die Lagereglerentwicklung zu beschränken, kann der Funktionsumfang über Plugins erweitert werden. Diese verknüpfen den Modellcode und den Rahmencode über entsprechende Schnittstellen miteinander.

Verwendete Toolboxen / Software

Derzeit ist zur Nutzung auf Sinumerik 840D Steuerungen der Siemens AG die Software NCKOEM der Siemens AG notwendig. Weiterhin werden folgende Toolboxen verwendet:

- Control System Toolbox
- Simulink Coder

Förderhinweis

Dieses Projekt wurde mit Mitteln aus dem Forschungs- und Innovationsprogramm Horizon 2020 der Europäischen Union im Rahmen der Finanzhilfevereinbarung Nummer 723440 finanziert.

Prof. Dr. C. Brecher / Arndt Wetzel, M.Sc.
Maschinendatenanalyse und NC-Technik
Lehrstuhl für Werkzeugmaschinen
Telefon: +49 241 80-28484
Mail: a.wetzel@wzl.rwth-aachen.de

Werkzeugmaschinenlabor WZL der RWTH Aachen University
Steinbachstraße 19 | 52074 Aachen | GERMANY
Telefon: +49 241 80-27400
Mail: info@wzl.rwth-aachen.de
Web: www.wzl.rwth-aachen.de

AUTOMATIC LON-BUS MODEL GENERATION USING SIMEVENTS

MATLAB Expo 2019, Munich, July 2th 2019

Alexander Martens, Christian Bock, Olaf Simanski, Olaf Hagendorf

Hochschule Wismar - University of Applied Sciences:
Technology, Business and Design
Research Group Computational Engineering
and Automation (CEA); www.cea-wismar.de
Contact: alexander.martens@hs-wismar.de

Lüftung
Klima & Kälte
Reinraumtechnik

EUROPÄISCHE UNION
Europäischer Fonds für
Regionale Entwicklung

Mecklenburg Vorpommern
MV tut gut.

IN SHORT …

Sensors, actuators and control devices can be connected e.g. via LON (Local Operating Network) bus. After installing a large network, communication problems can occur due to the number of communicating nodes, the data point types and the communication forms, as well as the cable length. One reason of this are difficulties of an error rate and busload estimation during project planning phase.
The Dr. Diestel GmbH is active in the field of ventilation, air-conditioning, refrigeration and clean room technology. To improve in their respective field of work, research is done, in the area of control and simulation of ventilation systems.
In order to plan a project more precisely, communication is simulated. This gives estimations about the reliability. Therefore, it can be decided, for example, to split a communication line into two individual lines or to change the device parameters.
This poster is concerning with automatic simulation model (SM) generation. Based on these models, evaluation of bus load and error rate estimation in LON networks is done. The automatic SM generation is based on the System Entity Structure (SES) and Model Base (MB) Framework. To meet the specific requirements, a template extension of the SES/MB Framework is introduced.

Project funded by: TBI MV, support code: TBI-V-1-135-VBW-048

LON-Bus –SES/MTB FRAMEWORK

SES is used to describe the multitude of systems permutations within a single tree structure. This includes the representation of variants for example a task, work steps, objects structure or SMs. After generate the SES, they can be pruned. The elements that are not needed are cut off. And the SES becomes a Pruned Entity Structure (PES). Each element of an SES has attributes. The attributes must be individual parameterize.
The SES/MB Framework [1,2,3] describes the pruning of desired system configurations from an SES and the generation of SMs using a MB. For transparent parameterization, the SES/MB Framework was extended with templates. A template consists of a parameter set applicable to several nodes simultaneously. Figure 1 represents the generation of a PES out of a SES up to the generation of a SM using the model and template base (MTB).

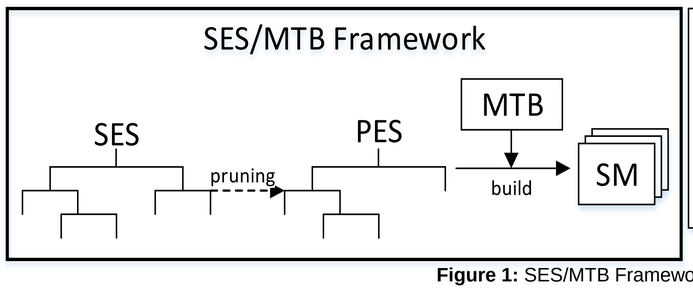


Figure 1: SES/MTB Framework

LON-BUS MODEL GENERATION

The simulation of LON bus models are realized with SimEvents (Version 4.4.1). They are automatically generated in five steps by using the SES/MTB Framework (see Figure 2).

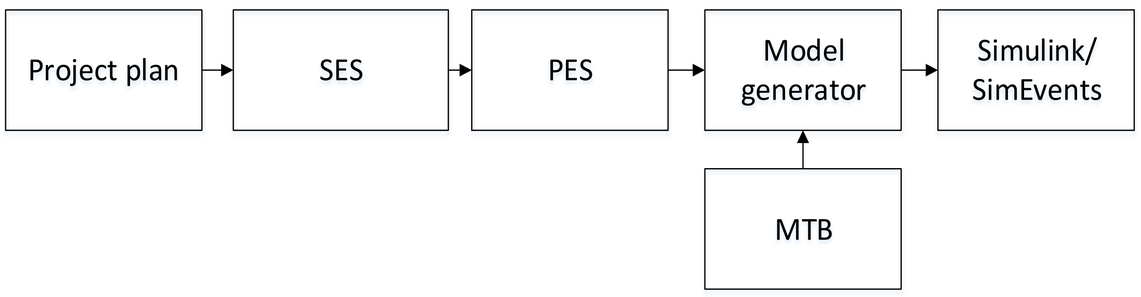


Figure 2: Overview Model Generation

In the first step all needed devices are drawn and connected in the project planning tool yEd-graph. Figure 3 shows a shortened line of an LON bus.

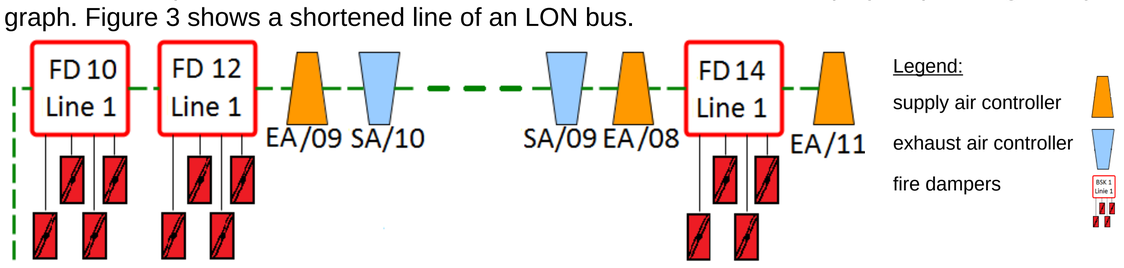


Figure 3: Example yEd-Graph with three device types (excerpt)

A project is saved as an XML file. With this, the structure of the LON bus is converted into a CSV file (comma separate values). The CSV file is manually extended by cable lengths, templates and the communication routes to other LON nodes (see Figure 4). The cable lengths between the LON nodes are estimated from the project documentation. The templates describe the function and properties of the individual LON nodes.

Line 1 UG LINX Port				
Cable from Device	to	Cable length	Template from column	Communication to line
Linx	FD/10	4m		
FD/10	FD/12	15m	Template_05	{Linx}
FD/12	EA/09	4m	Template_05	{Linx}
EA/09	SA/10	6m	Template_07	{SA/10, Linx}
SA/10	…	4m	Template_06	{EA/09, Linx}
…	…	…	…	…
…	SA/09	4m	Template_03	{SA/09}
SA/09	EA/08	8m	Template_02	{EA/11}
EA/08	FD/14	3m	Template_01	{SA/09, Linx}
FD/14	EA/11	25m	Template_04	{Linx}
EA/11	0	8m	Template_07	{SA/09, Linx}
		109m		

Figure 4: List of cable lengths between the LON nodes (excerpt)

With this information, the SES is generated and the PES is derived. After this, an SimEvents model is generated with the Simulink LON library [4]. Every LON model consist of LON nodes (MAC-Controller, one or more data points), T-Junction, line segments (Wires) and two terminators. The MAC-Controller implements the LON bus physical layers, a predictive p-persistent CSMA algorithm. It acts according to the LonTalk protocol. The data points generate DPs (data packets) at previously set time intervals with or without acknowledgment. T-Junctions are used to connect LON devices to the bus. The line segments, which means the associated length parameter, cause a proportional time delay of transmitted DPs. The terminators delete DPs, at the end of the bus.
Figure 5 shows a generated Simulink model (shortened) with all LON bus components. The shown model consists of four LON nodes from figure 3 with their data points. The data points and line segments are parameterized according to given templates and the cable length.

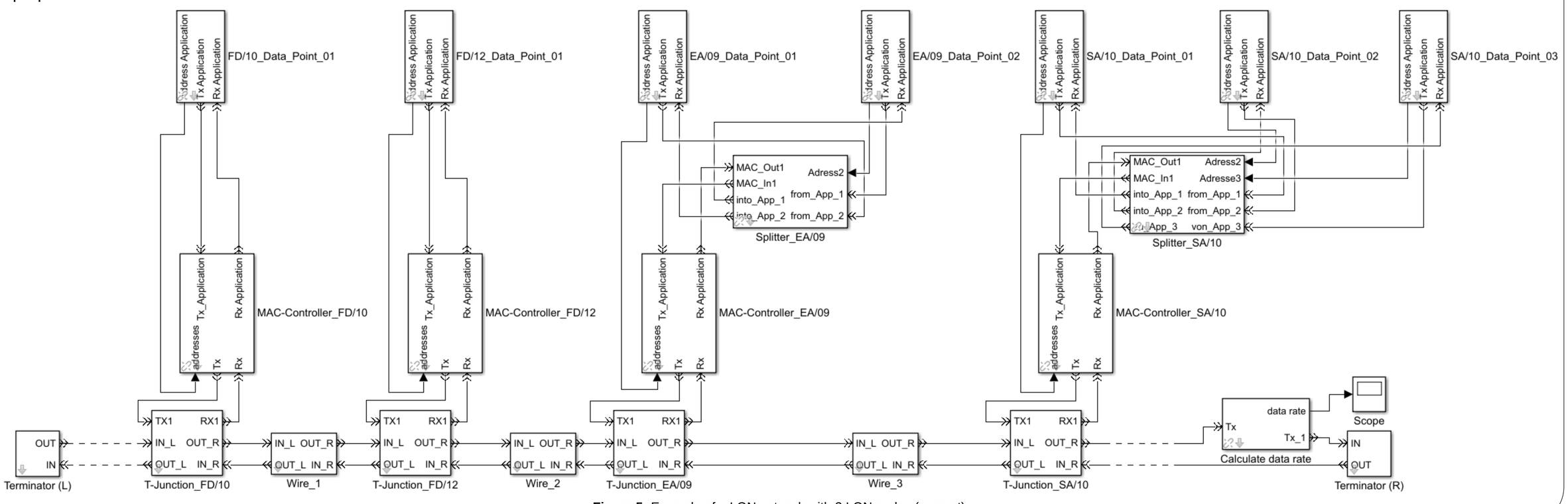


Figure 5: Example of a LON network with 3 LON nodes (excerpt)

CONCLUSION

The implementation of a SES/MTB based model generation is using information from the project planning. It enables an early project validation. This makes it possible to analyze network busloads and error rates already during the project planning phase. At the beginning, the user adds the required devices to a LON bus line using a project planning tool and estimates the necessary cable lengths. The user then determines applicable templates for respective LON nodes. This information is used for the automatic SES and modell generation and subsequent simulation. The user can identify potential network problems using the simulation results. He can manually change network structure or templates and repeat generation/simulation to improve the reliability.

Reference:
[1] Zeigler B. P, Praehofer H, Kim T. G., Theory of Modeling and Simulation. Academic Press 2000
[2] Hagendorf O, Simulation Based Parameter and Structure Optimisation of Discrete Event Systems. ARGESIM/ASIM – Verlag, Vienna, 2010
[3] Deatcu C, Folkerts H, Pawletta T, Durak U, How to Define SES Trees for Variability Modeling. ASIM 2018 – 24. Symposium Simulationstechnik, 2018 October, Germany, pp. 35-44
[4] Martens A, Bock C, Simanski O, Hagendorf O, Automatische Modellbildung mittels SES/MB Framework und einer Template-Erweiterung, 2019 February, Germany

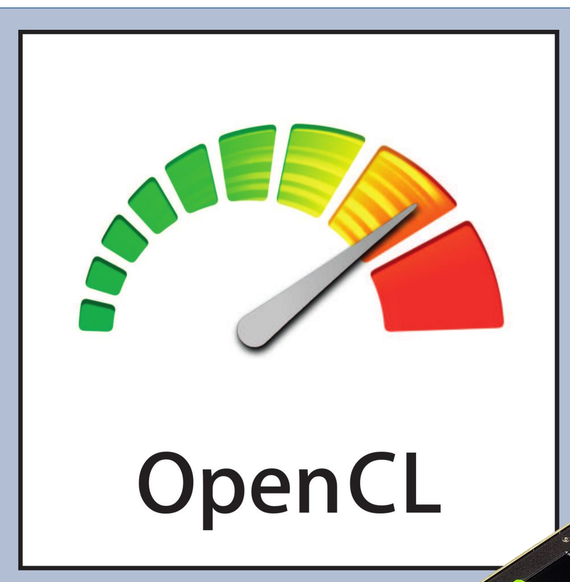
IANW Institut für angewandte numerische Wissenschaft e. V.

MatCL - Eine einfach anzuwendende OpenCL Toolbox für MATLAB

Philip Heinisch, Katharina Ostaszewski, Hendrik Ranocha

MatCL Toolbox

- MEX basierte Toolbox für OpenCL
- Direktes Ausführen von OpenCL Kerneln
- Kapselt OpenCL API vollständig
- Automatische Speicherverwaltung
- Erkennt automatisch MATLAB Datentypen
- Integrierte Laufzeitmessung
- Umleitung von OpenCL Meldungen (z.B. printf) zur MATLAB Konsole
- OpenCL Tests & Entwicklung
- Beschleunigung von Simulationen und Modellrechnungen
- Unterstützt MATLAB und Simulink
- Windows, Linux, macOS
- Quelloffenes GitHub Projekt: https://github.com/IANW-Projects/MatCL



OpenCL (Open Computing Language)

- Schnittstelle für heterogene Parallelsysteme
- Programme als portable OpenCL C Kernel
- Unterstützt CPU, GPU, ARM, FPGA, DSP
- Verwaltet durch Khronos Group (OpenGL/Vulkan)



API Beispiele

OpenCL Geräte auflisten

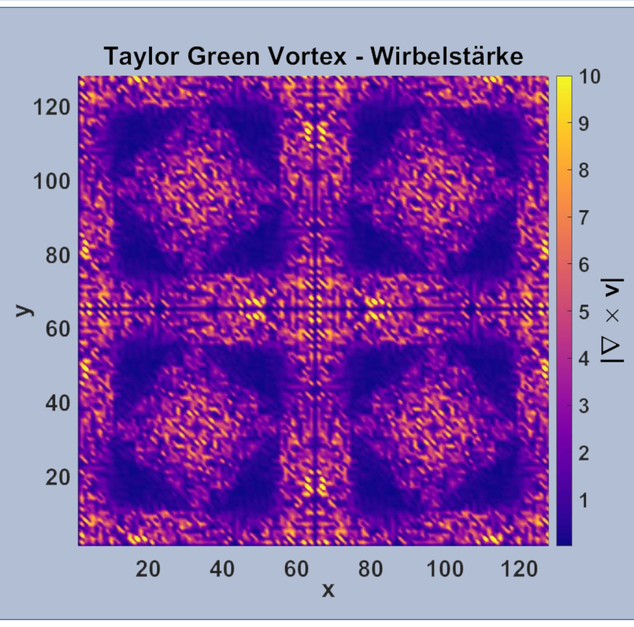
```
[names,dev_class,max_mem,max_wgsize,max_lwsize,compute_units]=cl_get_devices();
```

Kernel kompilieren und ausführen

```
[run_time]=cl_run_kernel(ocl_dev_id,'kernel.cl ','defines', 'cl_func',grange,lrange,in1…,out1…,[rw_flags]);
```

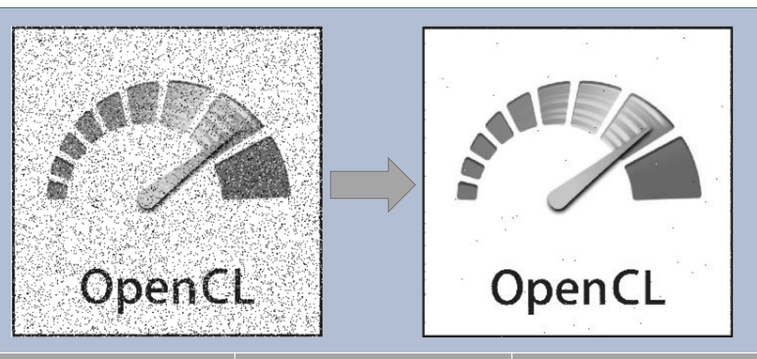
Anwendung: Testen & Entwickeln von OpenCL basierten Simulationen

- Integration in MATLAB erlaubt einfaches Testen & Validieren von OpenCL Algorithmen
- Interaktive Entwicklung von OpenCL Kerneln zum späteren Einsatz auf Clustern, FPGAs etc.
- Anbindung von OpenCL Modulen an MATLAB, z.B. zum Postprocessing
- Generieren von Testdaten mit MATLAB erleichtert Optimierung und Debugging von OpenCL Kerneln



Anwendung: 2D Median Filter

- z.B. Kantenerkennung oder Rauschunterdrückung
- Problemgröße 4096 x 4096 (uint8)
- Direkter Vergleich: MATLAB, CUDA, MatCL-OpenCL, OpenMP (C++)



	MATLAB - CPU	MATLAB - CUDA	OpenMP - CPU	MatCL - CPU	MatCL - GPU	MatCL - iGP
Core i7-8750H GTX1050Ti/UHD 630	24 ms	9 ms	28 ms	15 ms	18 ms	27 ms
Ryzen 5 1600X GTX1060	23 ms	8 ms	22 ms	13 ms	17 ms	-

continuity
ELECTRONICS MADE SIMPLE

We Provide Solutions for electronics engineering

Who are we?

Continuity is the future of electronics engineering. Schematics, PCB layout and software at the push of a button. Half the time, half the costs, or even less.

Continuity is providing a full-stack model-based design approach for embedded systems. This means integrating embedded hardware and software design into one coherent model.

From that master model, domain specific models for software and hardware is derived.

Problems of electronics engineering:

Complex

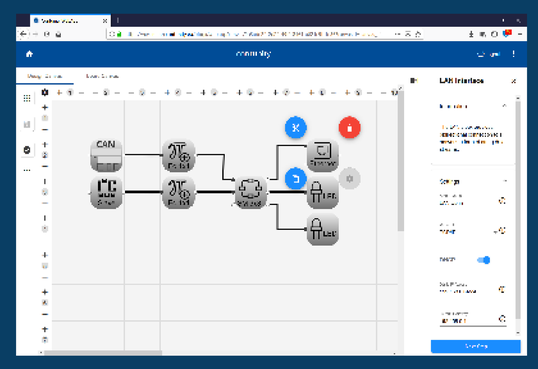
High cost

High Effort

Competence

Our Solution:

Design a simple block diagram



How our software algorithms work:

- Continuity provides full system models for hardware and software of embedded systems
- using MATLAB Simulink, Embedded Software submodels are derived
- Embedded Software submodel is combined with HAL libraries using 3rd party MATLAB Simulink toolboxes
- MATLAB scripting is used to reach full automation of the process
- with Embedded Coder the C-code is generated.

Our algorithms do the work

Hardware
(proprietary backend)

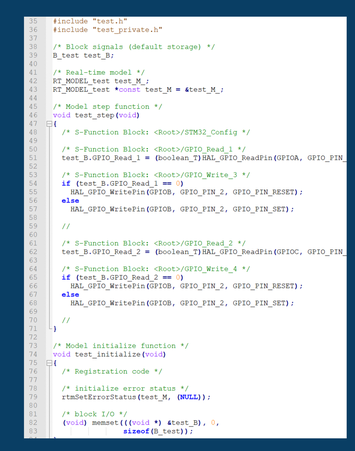
Software
(MATLAB Simulink)

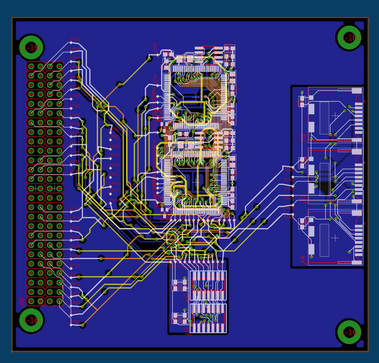
Why do we use MATLAB Simulink?

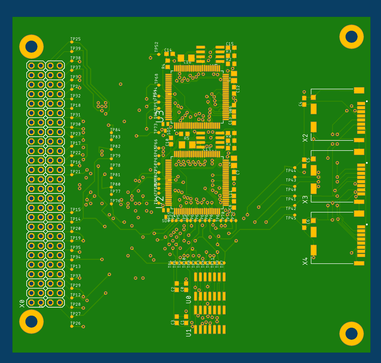
Mathworks MATLAB Simulink tool-chain with Embedded Coder, Matlab Coder, and many other extensions and toolboxes plays an important role in our product development. Especially for generating hardware independent embedded source code, these tools provide highest value to Continuity's customers. Together with Continuity's proprietary model-based hardware automation backend, Embedded Systems design can be accelerated up to 10 times, with 90% cost reduction.

Moreover, the synergies between Continuity's software platform and MATLAB Simulink allows reducing know-how barriers, through automation and simplification. This enables entire industries, to secure their competitive advantage throughout disruptions like digitalization and Industrial IoT.

Enjoy the outcomes: layouts, schematics, BOM, SW code







Contact us!

Lichtenbergstr. 8, 85748 Garching bei München
info@continuity.eu | www.continuity.eu | +49 (0) 89 125 013 81-0

Analog co-simulation of a RLC- Circuit with MATLAB/Simulink and Cadence ADE

M. Heitauer, M. Versen
Technical University of Applied Sciences Rosenheim, Rosenheim, Bayern

Technische Hochschule **Rosenheim**

Abstract

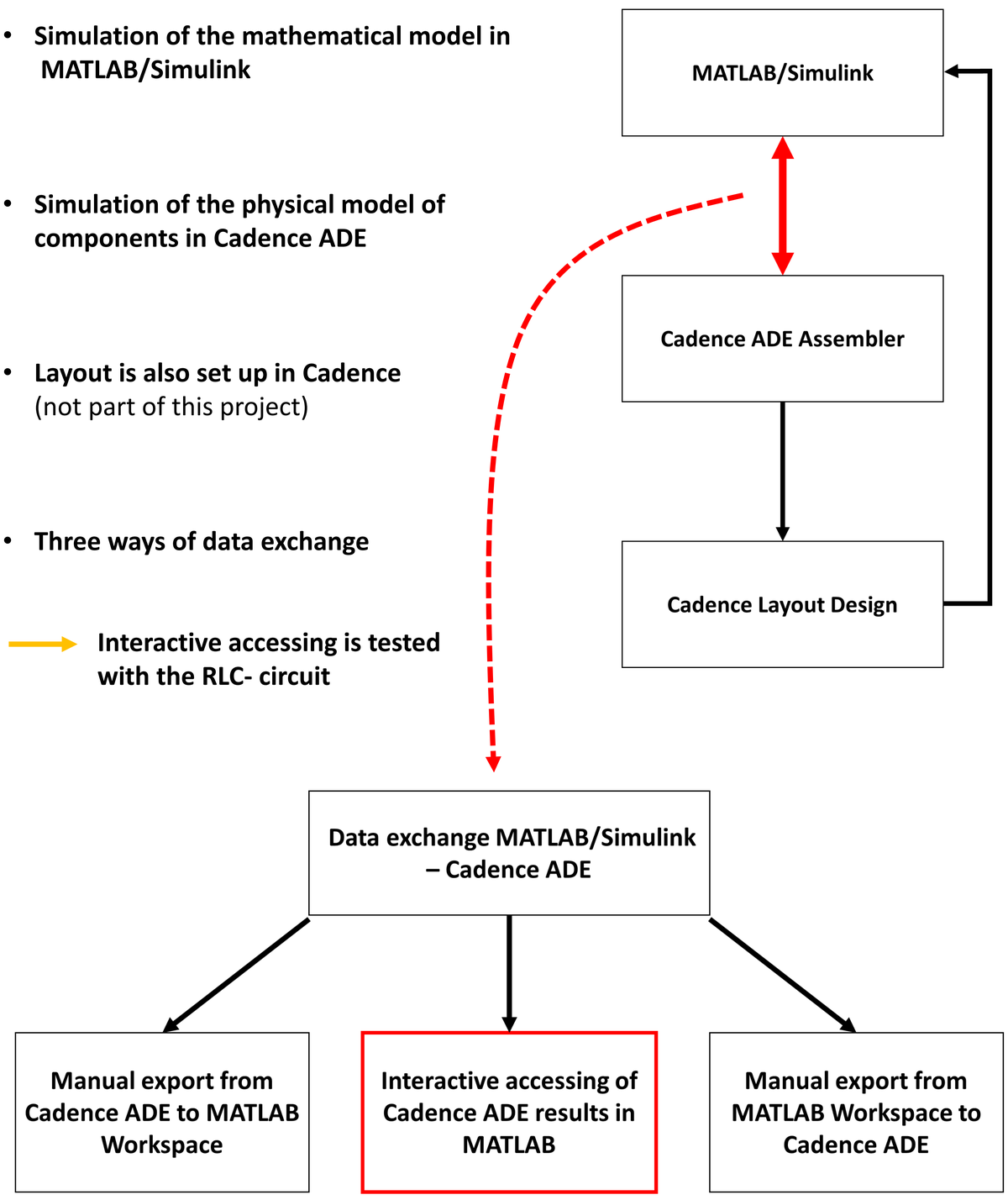
- In the project a co-simulation of a serial RLC- circuit is realized.
- The functions used are introduced in the *rapid adoption kit* by Cadence.
- After the simulation in Cadence ADE, the simulation of the MATLAB/Simulink model starts automatically and the results of both simulations are compared with an appropriate MATLAB- function.
- The bandpass curve of the circuit is simulated with both models and the results are compared.

Steps of the co-simulation

The objective of the project is to automatically co-simulate the bandpass behavior of the serial RLC- circuit with MATLAB/Simulink and Cadence ADE Assembler

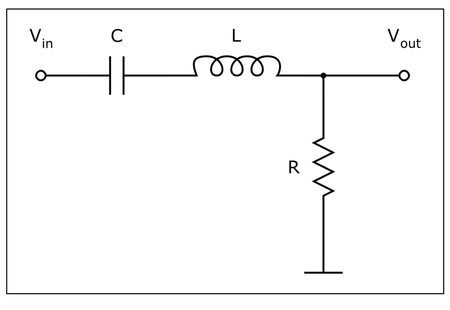
- Simulation of the mathematical model in MATLAB/Simulink
- Simulation of the physical model of components in Cadence ADE
- Layout is also set up in Cadence (not part of this project)
- Three ways of data exchange

→ Interactive accessing is tested with the RLC- circuit

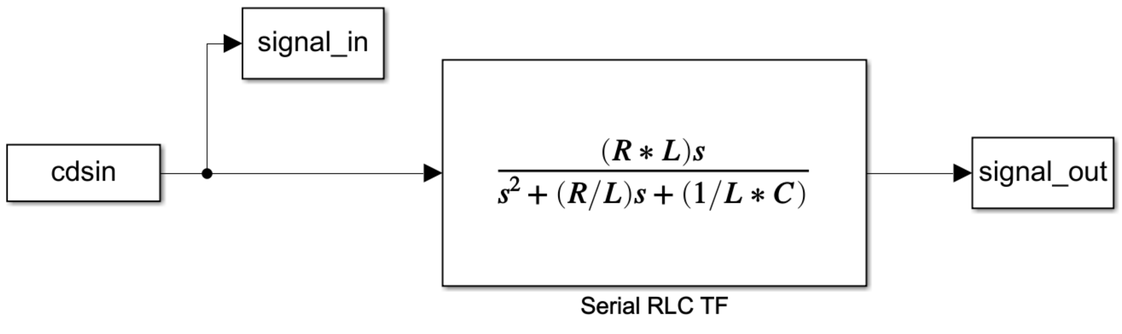


MATLAB/Simulink model

- Model of shown RLC- circuit is realized with transfer function block
- Transfer function: $H(s) = \frac{V_{out}}{V_{in}} = \frac{\frac{R}{L}s}{s^2+\frac{R}{L}s+\frac{1}{LC}}$
- Input signal *cdsin* is used from Cadence ADE

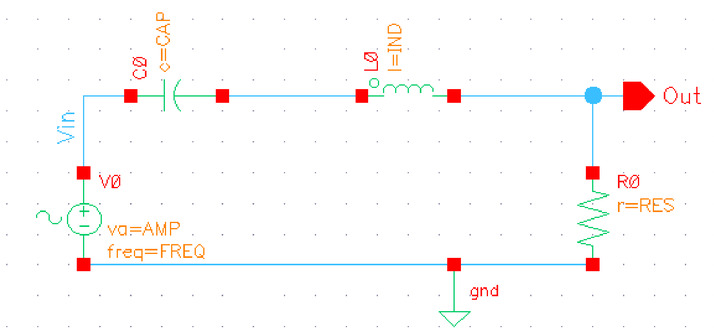


- Input and output signal are written to workspace (WS)



Simulation setup in Cadence ADE

- Model with components from analog standard lib.
- Design variables for values of resistance, capacity etc.
- Type of simulation as well as stepsize and endtime are set.



- Signals for input and output voltage defined as outputs of the simulation:

RLC_Test	Vout	signal	/Out	point
RLC_Test	Vin	signal	/Vin	point
RLC_Test	Bandpassplot	matlabexpr	rlcsim()	all

→ After all simulations are finished, the third output of the type *matlabexpr* is evaluated.
→ MATLAB is started and the function *rlcsim()* is run automatically.

Evaluation of the MATLAB- function

Load ADE simulation data

↓

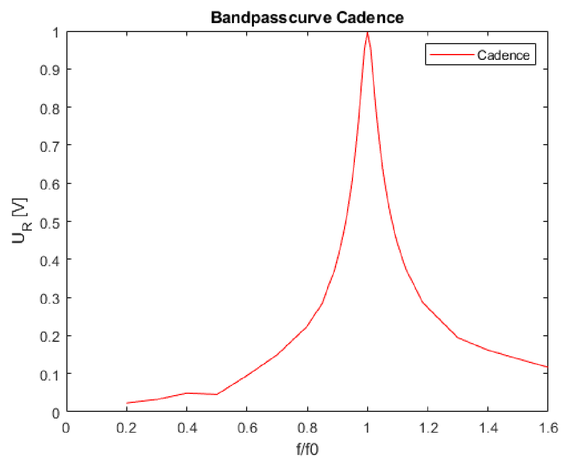
Store design variables to WS
Store outputs of ADE simulation to WS
Create three vectors for bandpass plot (frequency,vout,vout-simulink)

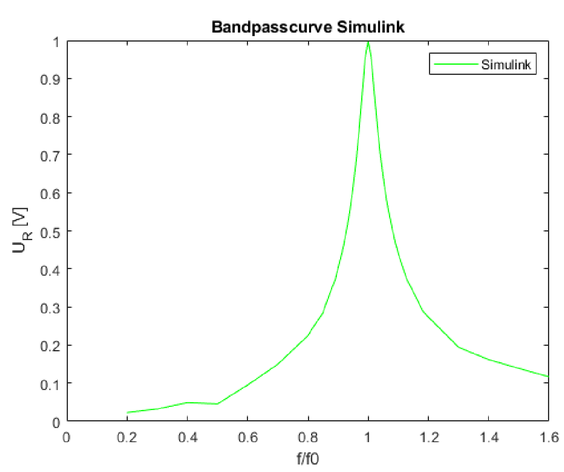
↓

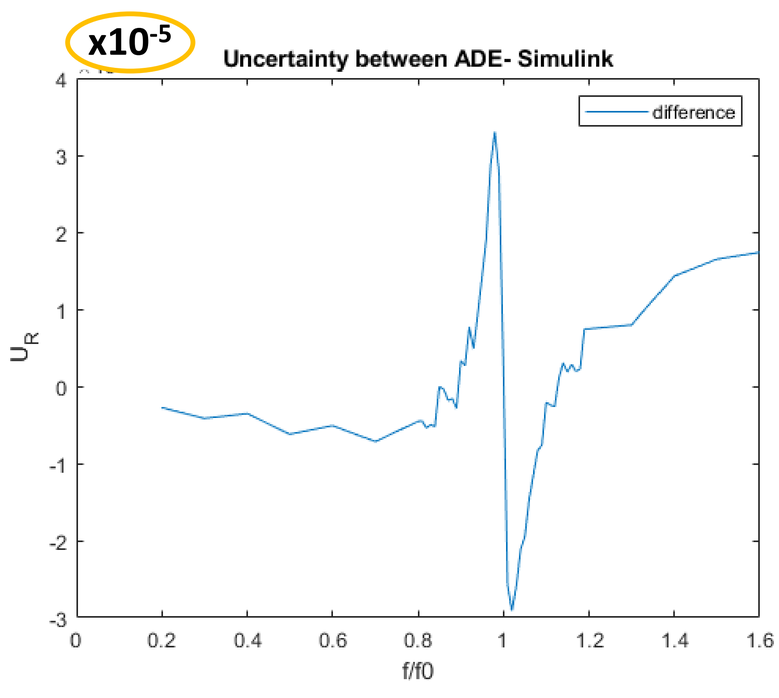
for each frequency point:
Simulate Simulink model
Fill vectors for bandpass plot
Calculate difference of ADE and Simulink

↓

Calculate resonance frequency f_0
Plot bandpass curve over normalized freq.
Plot difference in curves over normalized freq.







Summary

- Co-simulation was successful.
- Cadence ADE and Simulink lead to the same results.
- MATLAB enables fast handling of Cadence ADE results.

→ Next step: Simulation of a MOSFET transistor circuit

Rupprecht-Gymnasium München
Naturwissenschaftlichtechnologisches
und Sprachliches Gymnasium
ZUSAMMEN.WACHSEN.

Price Setting in monopolistic and oligopolistic markets

Tommy Duong

Introduction

- Economics is a highly complex topic. There are a countless amount of variables that have to be considered if one wants to understand the mathematics behind the various markets. Diving deeper into the different kinds of markets the most common ones are the monopoly, polypoly and the oligopoly. Starting with the monopoly, there is excatly one supplier who is facing a numerous amount of buyers. The mathematics behind this kind of market are quite simple and easy to understand.
- Moving further to the oligopoly, the by far most interresting kind of market, overall you can say that there are many suppliers and a small amount of buyers.
- With the polypoly oustandig, it is a Market situation where there are large number of small buyers and small sellers, none of which can influence prices.
- This is just a very small amount of what economics can offer. Since there are so many variables, which constantly change, the entire topic of economy is very interresting and alot of possible research which could be done.[1]

Theoretical Background

- As already mentioned above there are a numerous amount of variables to be considered before you can make a proper calculation regarding setting the right price. In my project I will solely focus on the mathematical aspect and will not discuss other things such as moral and ethics.

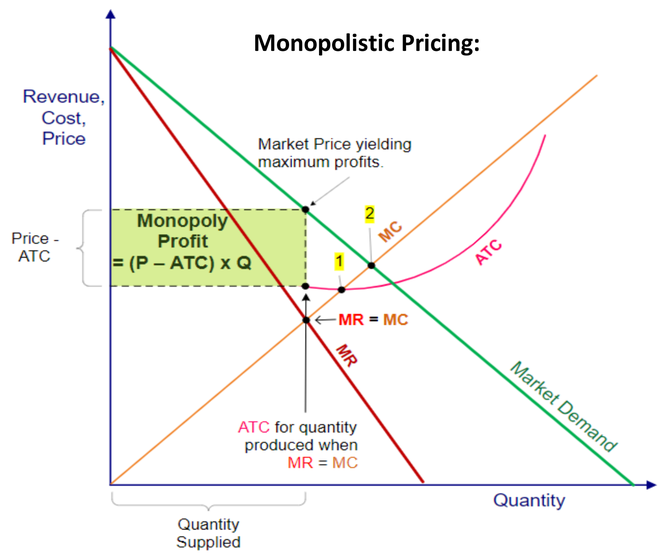


Figure 1 [3]

Theoretical Background

- In monopolies the goal of the seller is to get an equal marginal revenue as marginal cost because as long as marginal revenue exceeds marginal cost, then producing an additional unit will increase profits. When marginal revenue equals marginal cost, then the monopolist looks to the demand curve to see what price that corresponds to. At that point, profit is maximized. If the monopolist increases production beyond MR = MC, then the marginal cost will be greater for each additional unit than marginal revenue, which will decrease profits, since costs continue to increase.[2]
- The oligopoly is a market structure in which there are a few firms producing a product. When there are few firms in the market, they may collude to set a price or output level for the market in order to maximize industry profits. As a result, price will be higher than the market-clearing price, and output is likely to be lower. At the extreme, the colluding firms may act as a monopoly, reducing their individual output so that their collective output would equal that of a monopolist, allowing them to earn higher profits.

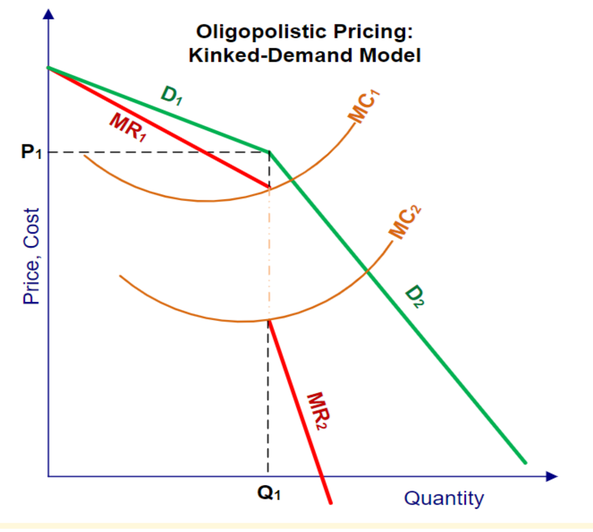


Figure 2 [4]

Theoretical Background

- $\mathbf{P_1}$ = Product Price of the Oligopoly
- If a firm raises its price ($\mathbf{D_1}$), but the others do not match the increase, then revenue will decline in spite of the price increase.
- If the firm lowers its price (**D2**), then the other firms will match the decrease to avoid losing market share.
- Because there is a kink in the demand curve, there is a gap in the marginal revenue curve ($\mathbf{MR_1}$ - $\mathbf{MR_2}$). Since firms maximize profit by producing that quantity where marginal cost = marginal revenue, the firms will not change the price of their product as long as the marginal cost is between $\mathbf{MC_1}$ and $\mathbf{MC_2}$, which explains why oligopolistic firms change prices less frequently than firms operating under other market models.[2]

Implementation in MATLAB

- In my further progress I will try to create MATLAB scripts based on the knowledge that I gained regarding the mechanics of both monopolies and oligopolies that will calculate the ideal price to set as a seller.
- I will start with the monopoly since I expect oligopolies will be more difficult to implement. My goal is to have two working scripts or if possible one that will give an exact price to set for the seller after providing the mandatory data

References:
[1] Jerry Green: *Microeconomic Theory*. Oxford University Press, Oxford 1995, ISBN 0-19-507340-1, S. 385
[2] William C. Spaulding:https://thismatter.com/economics/pure-monopoly-demand-revenue-costs-profits.htm (03.06.2019)
[3] figure 1:https://thismatter.com/economics/pure-monopoly-demand-revenue-costs-profits.htm (03.06.2019)
[4] figure 2:https://thismatter.com/economics/oligopoly-pricing-models.htm (03.06.2019)

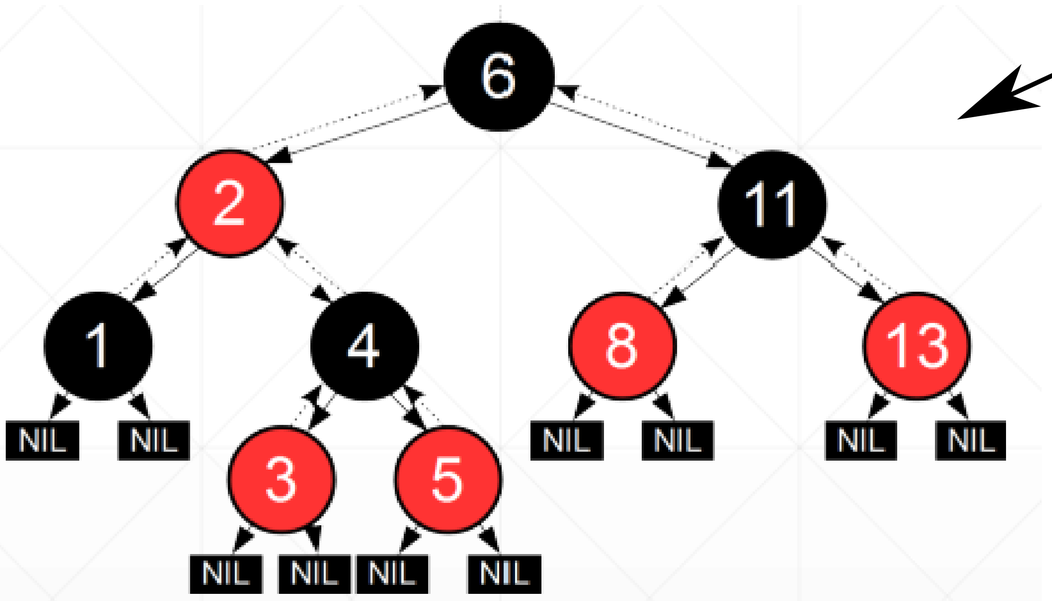
Rupprecht-Gymnasium München
Albrechtstraße 7
80636 München
mint@rupprecht-gymnasium.de

MathWorks
Adalperostr. 45, 85737 Ismaning
www.mathworks.de

MINT-EC
Poststraße 4/5, 10178 Berlin
info@mint-ec.de

Rot-Schwarz-Baum

Ein balancierter binärer Suchbaum (engl.: „self-balancing binary search tree“)



b) Eigenschaften

- Bezüglich schwarzer Knoten optimal balanciert
- Jeder Unterbaum ist (abgesehen von Kriterium 1) ebenfalls ein RBT.
- Relativ ausgeglichene Pfadlänge

$$l_{max} \leq 2 \cdot l_{min}$$

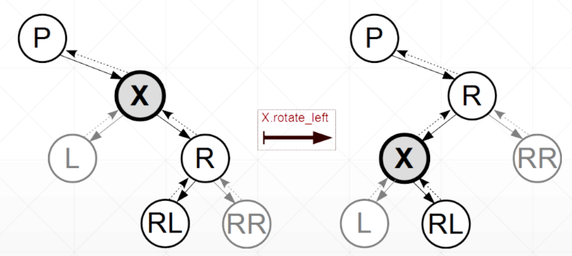
- Die Höhe steigt logarithmisch mit Anzahl der Knoten.

$$h \leq 2 \cdot {}_2\log(n+2) - 2 \leq c \cdot \ln(n)$$

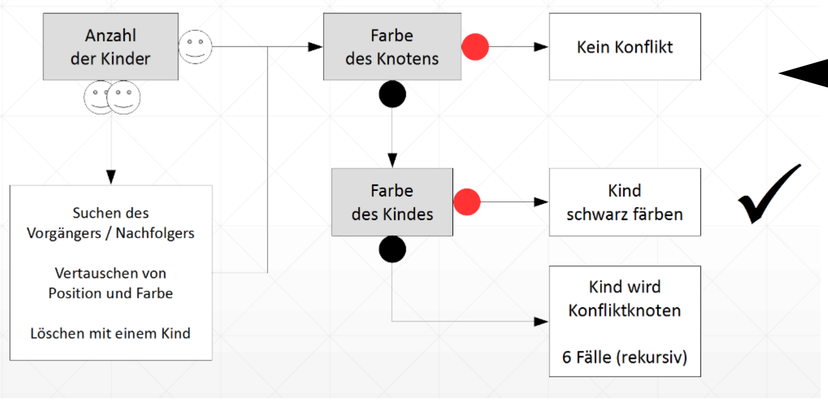
a) Kriterien

1. Die Wurzel ist schwarz.
2. Jedes Blatt (NIL-Knoten) ist schwarz.
3. Ist ein Knoten rot,
   so sind seine beiden Kinder schwarz.
4. Konstante Schwarztiefe

c) Realisierung | Rotationen



Realisierung | LÖSCHEN



c) Realisierung | EINFÜGEN

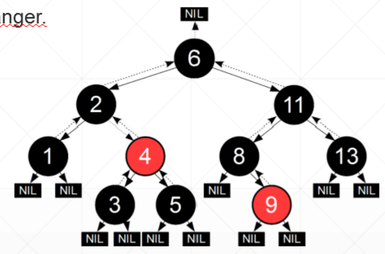
1. Erster Knoten im Baum => Knoten schwarz färben ✓
2. Elternknoten schwarz => Knoten rot färben ✓
3. Elternknoten rot => mind. 1 Kriterium nicht erfüllbar
   - Reparaturen notwendig
     Rotationen und Umfärbungen (teilweise rekursiv)
   - Drei weitere Fälle (3.a, 3.b, 3.c)
     bzgl. Einfärbungen diverser anderer Knoten

c) Realisierung | VORGÄNGER

(1) Der Knoten ist das MINIMUM und hat keinen Vorgänger.

(2) Der Vorgänger befindet sich

- ganz rechts im linken Unterbaum.
- auf dem direkten Pfad zur Wurzel.
  Der Vorgänger ist dann einer der Ahnenknoten.



Profile Summary
Generated 12-May-2019 16:50:28 using performance time.

Function Name	Calls	Total Time	Self Time*	Total Time Plot (dark band = self time)
RBTnode>RBTnode.insert_case3	1961	1.233 s	0.017 s	
RBTnode>RBTnode.uncle	2939	1.090 s	1.030 s	
initialize_RBT	1	0.493 s	0.012 s	
RBtree>RBtree.insert	1000	0.478 s	0.070 s	
RBtree>RBtree.find	1000	0.278 s	0.005 s	
RBTnode>RBTnode.insert_case4	983	0.116 s	0.030 s	
RBTnode>RBTnode.insert_case5	983	0.084 s	0.021 s	
RBTnode>RBTnode.isNIL	8893	0.058 s	0.058 s	
RBTnode>RBTnode.rotate_left	983	0.056 s	0.051 s	
RBTnode>RBTnode.grandparent	13722	0.026 s	0.026 s	
RBTnode>RBTnode.RBTnode	1001	0.020 s	0.010 s	
RBTnode>RBTnode.insert_case1	1978	0.016 s	0.011 s	
RBTnode>RBTnode.insert_case2	1969	0.013 s	0.008 s	
BSTnode>BSTnode.root	1000	0.010 s	0.004 s	
RBTnode>RBTnode.hasParent	1008	0.006 s	0.003 s	
BSTnode>BSTnode.BSTnode	1001	0.005 s	0.005 s	
RBTnode>RBTnode.dyeBlack	2949	0.005 s	0.005 s	
RBtree>RBtree.RBtree	1	0.004 s	0.001 s	
RBTnode>RBTnode.isRed	1961	0.003 s	0.003 s	
RBTnode>RBTnode.isBlack	1969	0.003 s	0.003 s	
RBTnode>RBTnode.dyeRed	1961	0.003 s	0.003 s	
RBTnode>RBTnode.repairNIL	1	0.001 s	0.001 s	

Performance || RS-Baum

Operation in logarithmischer Zeit:
Rechenzeit wächst logarithmisch mit Anzahl der Knoten

initialize_RBT.m | initialize_Matrix.m | demo_RBT.m | demo_Matrix.m

```
%% Erstellung eines Rot-Schwarz-Baums mit 1,000 Knoten
% Mit mit nicht unerheblich großer Datenmenge)

data = 1e9 .* ones(1e6, 1);
RBT = RBtree;
for k = 1:500
    RBT.insert(k, data);
end
for k = 506:1005
    RBT.insert(k, data);
end
```

```
%% RBT | Einfügen mit Schlüssel "501" bis "505"
for k = 501:505
    RBT.insert(k, data);
end
%% RBT | Einfügen am Ende
RBT.insert(1006, data);
```

Function Name	Calls	Total Time	Self Time*	Total Time Plot (dark band = self time)
demo_RBT	1	0.056 s	0.002 s	
RBtree>RBtree.insert	1	0.054 s	0.035 s	
RBtree>RBtree.find	1	0.011 s	0.001 s	
RBtree>RBtree.find0	11	0.010 s	0.010 s	
num2str	1	0.008 s	0.006 s	
int2str	1	0.002 s	0.002 s	

Performance || 2-dim array

Operation in linearer Zeit:
Rechenzeit wächst proportional zur Anzahl der Knoten

```
%% Erstellung eines 1,000,000 x 1,000 - Arrays (mit lauter Milliarden)
% Insbesondere im Binärformat sind Milliarden "nicht-trivial".
%
% Dies geschieht nicht durch Anfügen an die letzte Spalte mithilfe von
% horzcat(M,data), sondern Matlab-freundlich in einem Schritt.

data = 1e9 .* ones(1e6, 1);
M = 1e9 .* ones(1e6, 1e3);
center = 500;
```

Function Name	Calls	Total Time	Self Time*	Total Time Plot (dark band = self time)
initialize_Matrix	1	10.338 s	10.338 s	

Self time is the time spent in a function excluding the time spent in its child functions. Self time also includes overhead resulting from the process of profiling.

```
%% Matrix | Einfügen in der Mitte
for k = 1:5
    M = [ M(:,1:500), zeros(1e6,1), M(:,500+1:end) ];
end
%% Matrix | Einfügen am Ende
M = [M,data];
```

Function Name	Calls	Total Time	Self Time*	Total Time Plot (dark band = self time)
demo_Matrix	1	117.463 s	117.463 s	

Self time is the time spent in a function excluding the time spent in its child functions. Self time also includes overhead resulting from the process of profiling.

Compact use of functions on complex data structures

(Here: binary singleton expansion and array)

Plotting multiple lines using **bsxfun** and **arrayfun**

This example demonstrates the usage of the MATLAB® functions *bsxfun* and *arrayfun*. Code becomes shorter and easier to read.

Contents

- Create data
- Visualize data
- Add legend

Create data

```
row_E = linspace(-0.8,0.8,1000);
row_T = 0 : 0.025 : 0.2;
fun_fermi = @(E,T) ( 1 + exp(E/T) ).^(-1);
```

Execute bsxfun.

```
result = bsxfun(fun_fermi, row_E.', row_T);
```

No meshgrid, no loops, no pre-allocating. We are able to focus on the algorithm.

Visualize data

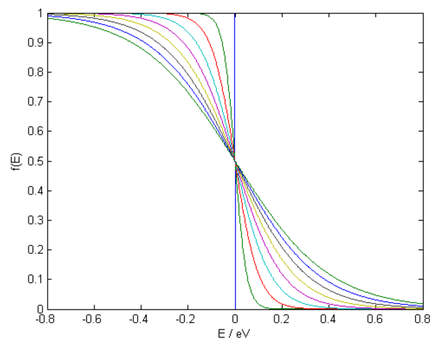
```
plot(row_E, result)
xlabel('E / eV')
ylabel('f(E)')
```

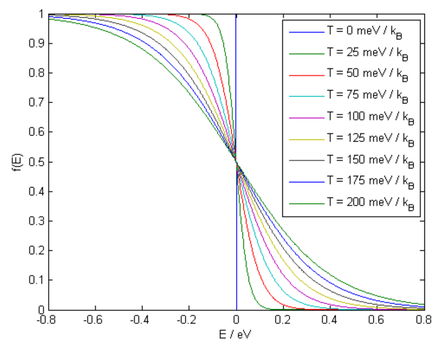
Add legend

```
fun_legstr = @(x) ['T = ',num2str(x*1000),' meV / k_B'];
```

Build legend string using *arrayfun* and abusing the *UniformOutput* flag in order to obtain a cell-array that can be handed over to legend.

```
legend( arrayfun(fun_legstr, row_T, 'UniformOutput', false) );
```





andreas.postl.email@gmail.com
DI Andreas Postl, Alumnus der Technischen Universität Graz, Doktoratsstudent (Physik) an der Universität Wien

Rupprecht-Gymnasium München
Naturwissenschaftlichtechnologisches und Sprachliches Gymnasium

ZUSAMMEN.WACHSEN.

Numerical Linear Algebra in Real World Applications

Anton Lin

Introduction

During the Gulf War in 1991, 28 soldiers died and around 100 other people were injured by failing to track and intercept an incoming missile (Arnold, 2000). This was caused by the accumulation of numerical errors using computers for time calculation. These numerical errors are due to rounding errors, since computers represent numbers only with finite precision using floating point arithmetic (today's standard: IEEE 754, 2008). Besides rounding errors, measurement errors and truncation errors can occur. This motivates the development of numerical methods and the numerical analysis of these problems. However this poster only deals with the study of numerical linear algebra, which means to develop numerical methods and analyze problems in the field of linear algebra and therefore is mainly employed to efficiently solve linear equation systems.



Source: http://www-users.math.umn.edu/~arnold//disasters/patriot.html (Accessed 8th June 2019)

Theoretical Background

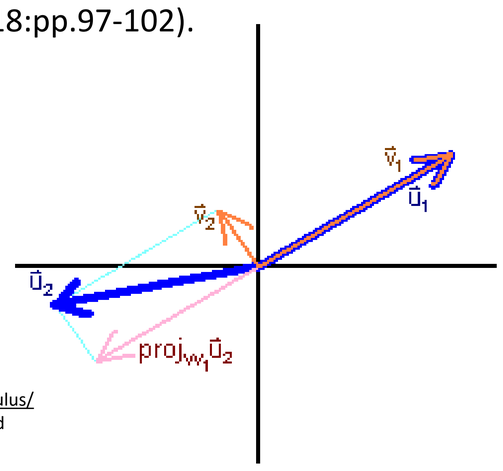
QR Decomposition

The QR decomposition splits a matrix $A \in \mathbb{R}^{m\times n}$, with $m \geq n$, into the orthogonal matrix $Q \in \mathbb{R}^{m\times n}$ and the upper triangular matrix $R \in \mathbb{R}^{n\times n}$ (Boyd and Vandenberghe, 2018:p.190).

$$A = QR \tag{1.1}$$

A possible approach to achieve (1.1) is using Gram-Schmidt orthogonalization (GS) to find orthonormal basis for the column vectors of A. It can be categorized in two different variations, the classical Gram-Schmidt orthogonalization (CGS) and the modified Gram-Schmidt orthogonalization (MGS), which changes the order of the steps (Boyd and Vandenberghe, 2018:pp.97-102).

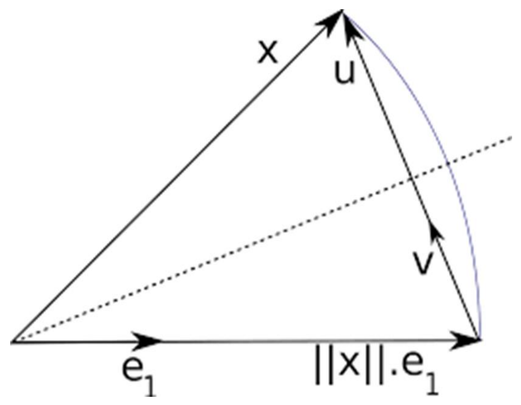
Visualizing the Gram-Schmidt orthogonalization



Source: https://www.math.hmc.edu/calculus/tutorials/gramschmidt/ (Accessed 8th June 2019)

Another method to solve for (1.1) is by using Householder triangularization (HHT), where the QR decomposition is acquired by applying Householder reflections onto the column vectors of A (Higham, 2002:pp.354-356).

Visualizing the Householder reflection



Source: https://commons.wikimedia.org/wiki/File:Householder.svg (Accessed 8th June 2019)

Least Squares

The Least Squares method is used to solve a overdetermined equation system for the coefficients c (Boyd and Vandenberghe, 2018:pp.225-231),

$$Ac = b \tag{1.2}$$

given $b \in \mathbb{R}^m$, such that the sum of the squared error $\| Ac - b \|^2$ is minimized, i.e.,

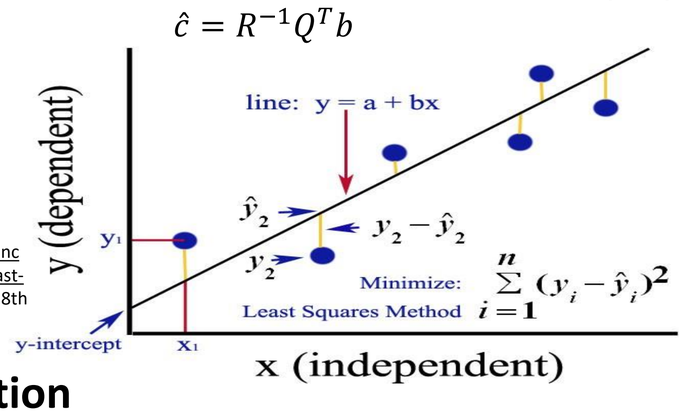
$$\hat{c} = \underset{c}{\operatorname{argmin}} \| Ac - b \|^2 \tag{1.3}$$

Solving (1.3) is equivalent to the solution of the normal equation

$$\hat{c} = (A^T A)^{-1} A^T b. \tag{1.4}$$

We can reformulate (1.4) using the QR decomposition such that

$$\hat{c} = R^{-1} Q^T b \tag{1.5}$$



Source: https://xaviergeerinck.com/ordinary-least-squares (Accessed 8th June 2019)

Simulation

We analyze CGS, MGS and HHT on an example, by using the various methods on a least squares problem, where samples of $f(x) = \sin\left(\frac{1}{x}\right)$ get approximated with n- th order polynomials. The horizontal axis represents inputs x, and $y = f(x)$ represents the vertical axis. The vector b consists of evaluations of $f(x)$ at 100 evenly spaced inputs between x=0.1 and x=3. The coefficients $\hat{c}$ of our polynomials are computed according to (1.5) and the errors ϵ are computed by $\epsilon = \| A\hat{c} - b \|^2$.

- In Fig. 1.1 the CGS, MGS and HHT graphs are identical with small fluctuations regarding the underlying function $f(x)$ with $\epsilon_{CGS} = \epsilon_{MGS} = \epsilon_{HHT} = 2.87832$.
- In Fig. 1.2 CGS has fluctuations ranging from -2 to 3. The graph of MGS fluctuates less compared to CGS before rapidly dropping at $x = 3$. Whereas the HHT approximation barely fluctuates and seems to be identical to the underlying function $f(x)$. The errors are ϵ_{CGS}=138.3562, ϵ_{MGS}=16.5975 and ϵ_{HHT}=0.0147. In this figure HHT approximates $f(x)$ best, whereas CGS shows the worst results.

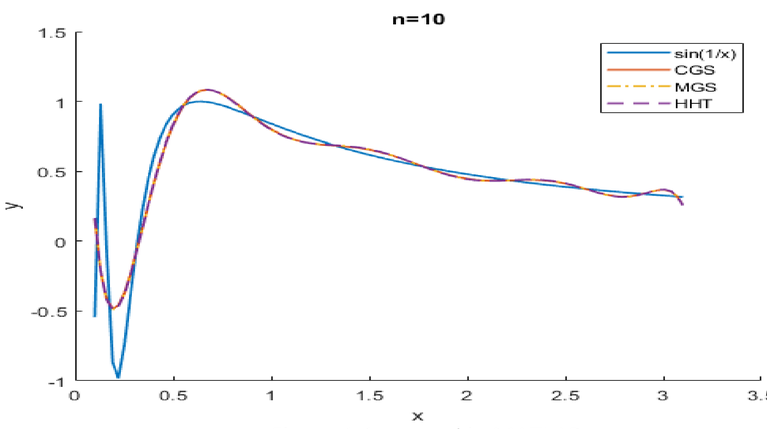


Figure 1.1 created in MATLAB

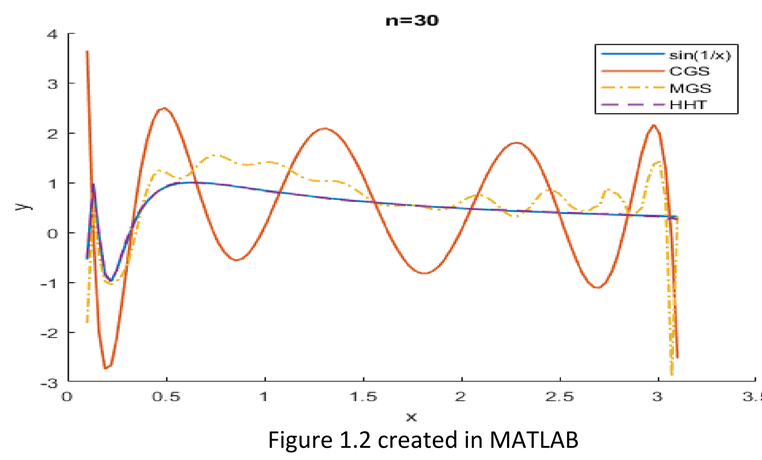


Figure 1.2 created in MATLAB

These quality differences using different algorithms are due to the numerical issues caused by finite precision arithmetic by using computers. Therefore, the computed Q matrices are not exactly orthogonal. A measure for orthogonality is given by o = $\| Q^T Q - \mathrm{I} \|$, which can be computed for all three approaches, where $\|\cdot\|$ denotes the *Frobenius* norm. For $n = 30$ the values for orthogonality are o_{CGS} = 20.4418, o_{MGS} = 2.3651 and o_{HHT} = 2.2212. The smaller the values of o , the better the orthogonality properties of Q, with o = 0 being the ideal case. The obtained values explain and match our observations.

From this data it can be concluded that HHT is the preferable algorithm for our example, since the approximations become less erroneous by increasing n, while the errors increase, when increasing n for the approximations of CGS and MGS (MGS less than CGS).

```
function [W,R] = HHT(A)

%calctulats the the QR-decomposition of a tall m x n
%matrix A by applying the Householder Traingularization
%And getting the upper triangular matrix R nad the matrix W
%that implicitly defines Q

n = size(A,2);
m=size(A,1);
%making R to A, since Q'*A=R
R = A;
%creates a matrix W which implicitly defines Q
W = zeros(size(A));
%applying the i-th reflection matrix on R
for i = 1:n
    %defining the part/vector x of the matrix R
    %that is significant for the step
    x=R(i:end,i);
   %applying the householder reflecion on x
   %the reflction matrix is defined as I-2*w*w'
    [w,r] = householder(x);
    W(i:end,i)=w;
    %applying the reflection matrix on the i:n-th
    %components of all collumns of R
     R(i:end,i)=[r;zeros(m-i,1)];
    for j=i+1:n
    y=R(i:end,j);
    R(i:end,j)=y-2*w*(w'*y);
    end
end
%only taking the neccesary parts of R
%by cutting out the unneccesary zeros
R=R(1:n,1:n);
end
```

MATLAB code for HHT

```
function [Q,R] = CGS (A)
%computes the [Q,R] by applying the
%Gram-Schmidt orthogonalization
n=size(A,2);
%creating the nxn  triangular matrix R
R=zeros(n);
%creating the mxn orthogonalmatrix Q
Q=zeros(size(A,1),n);
%calculating the i-th orthogonormal vector q
for i=1:n
    v=A(:,i);
    %creating and substracting the orhtogonal projections
    %of the previous vectors q
    for j=1:i-1
        %for the MGS A(:,i) is replaced by v
        R(j,i)=Q(:,j)'*A(:,i);
        v=v-R(j,i)*Q(:,j);
    end
    %normalizing the new vector q
    R(i,i)=norm(v);
    Q(:,i)=v/R(i,i);
end
end
```

MATLAB code for CGS/MGS

References:

- Arnold, D. N. (2000). The patriot missile failure. Available from: http://www-users.math.umn.edu/~arnold/disasters/patriot.html[Accessed 10th May 2019].
- Boyd, S. and Vandenberghe, L. (2018). Introduction to Applied Linear Algebra, United Kingdom, Cambrigde University Print. Available from: http://vmls-book.stanford.edu/vmls.pdf[Accessed 11th May 2019].
- Higham, N. (2002). Accuracy and Stability of Numerical Algorithms, second edn, United States, the Society for Industrial and Applied Mathematics. Available from: http://servidor.demec.ufpr.br/CFD/bibliografia/Higham_2002_Accuracy%20and%20Stability%20of%20Numerical%20Algorithms.pdf[Accessed 11th May 2019]
- Smith, S. W. (1999). The Scientist and Engineer's Guide to Digital Signal Processing, second edn, San Diego, California Technical Publishing.Available from: https://users.dimi.uniud.it/~antonio.dangelo/MMS/materials/Guide_to_Digital_Signal_Process.pdf[Accessed12th May2019].

Rupprecht-Gymnasium München
Albrechtstraße 7
80636 München
mint@rupprecht-gymnasium.de

MathWorks
Adalperostr. 45, 85737 Ismaning
www.mathworks.de

MINT-EC
Poststraße 4/5, 10178 Berlin
info@mint-ec.de

Nutzung des ADM-Modells zur Optimierung des Methanflusses in Biogasanlagen

David Wagner, Wolfgang Schlüter

- Biogasanlagen setzen organische Biomasse durch anaerobe Fermentation in Biogas (Methan als Wertprodukt) um
- Das komplexe Verfahren lässt sich aufgrund vieler Einflussgrößen nur numerisch optimieren
- Das Anaerobic-Digestion Model 1 (ADM-Modell) ist das detaillierteste Modell für den Prozess

1. Hintergründe

- Energiewende macht ein Umdenken in der Energieversorgung notwendig
- Saisonale Schwankungen bei Wind- und Sonnenenergie
- Biogasanlagen können, flexibel betrieben, Lastspitzen abfangen
- Voraussetzung:
  - Sie müssen schnell, möglichst viel Biogas liefern
- Betrieb der Anlagen weit jenseits des Optimums

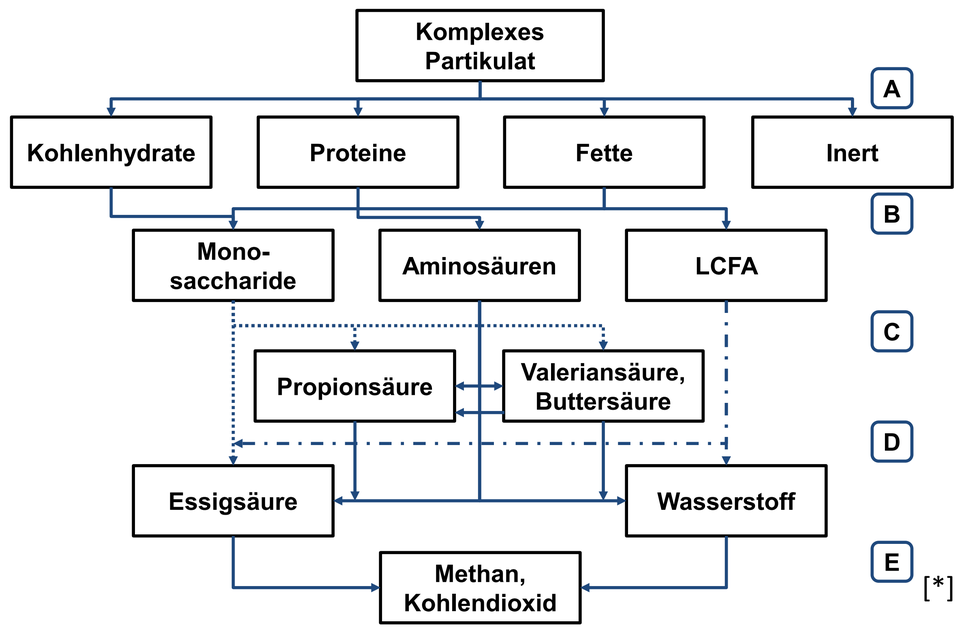
2. Ziele des Projekts

- Identifizierung unsensitiver Parameter
  - Ggf. Komplexitätsreduktion
- Evaluierung des Optimierungspotentials

3. Beschreibung des Modells

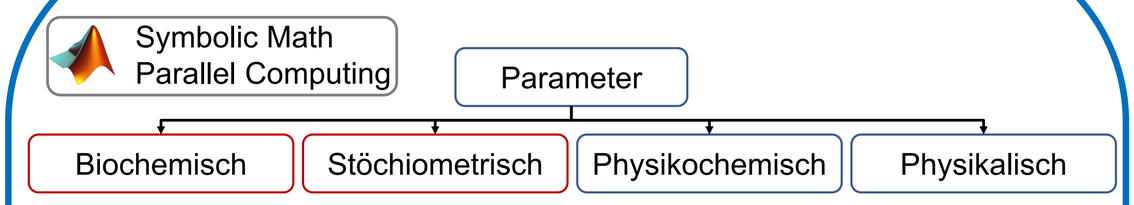
- 2002 veröffentlicht als Generalisierung der anaerober Digestionsmodelle
- Beschreibt physikochemische als auch biochemische Phänomene
- Insgesamt 39 ODEs und über 100 Parameter

4. Definition der Prozessschritte

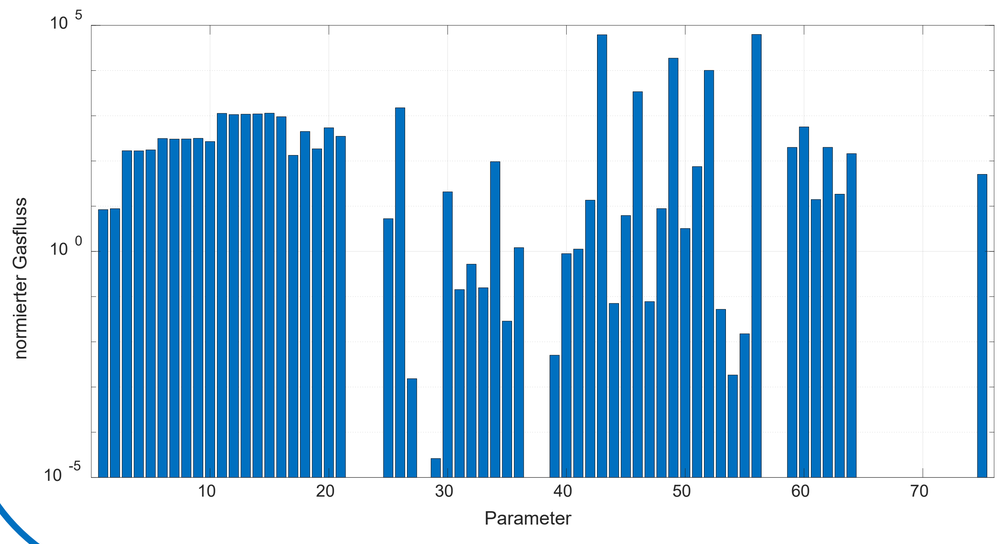


A – Desintegration
B – Hydrolyse
C – Versäuerung
D – Acetogenese
E – Methanogenese

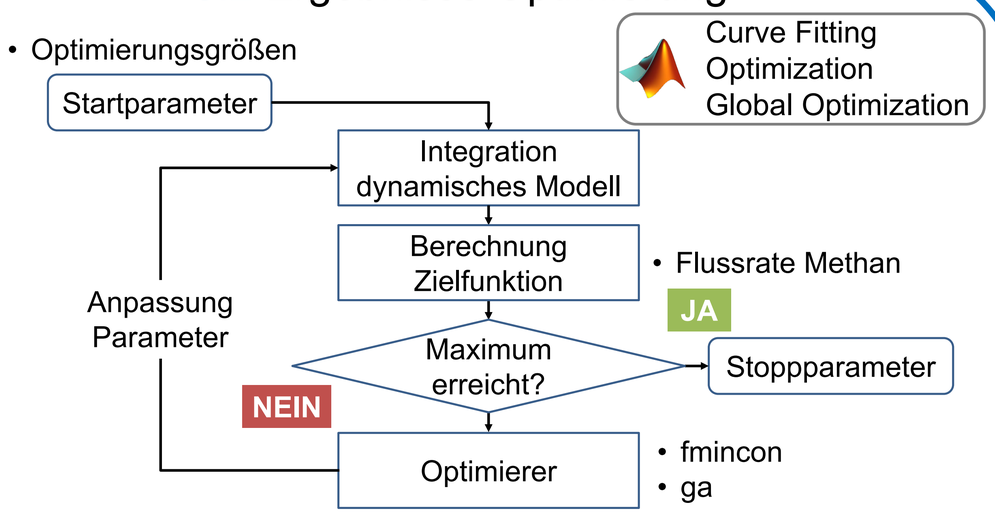
5.1 Ergebnisse Sensitivitäten



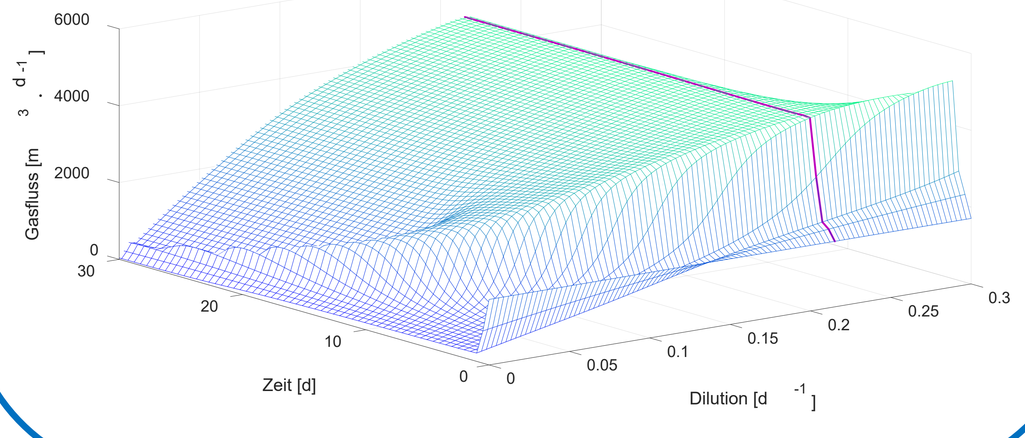
- Von 75 hier getesteten Parametern (biochemisch und stöchiometrisch) sind 17 unsensitiv
- Darunter Sterbekonstanten der Bakterien und zusätzlich in das Modell eingebundene Präzipitationskonstanten
- Die Ergebnisse stimmen mit der Literatur überein
  - Sterberaten der Bakterien sind vernachlässigbar, was besonders für den flexiblen Betrieb von Vorteil ist



5.2 Ergebnisse Optimierung



- Simultane Optimierung der Substratzusammensetzung und Dilutionsrate
- über 60%-ige Steigerung der Methanflussrate erreichbar (gegenüber Standardbedingungen)



6. Ausblick

Model Predictive Control

- Validierung der Modellparameter an Realanlagen
- Reduzierung des Modells im Hinblick auf Rechenzeit und Komplexität

→ Aufbau einer modellprädiktiven Regelung

verwendete Toolboxen

[*] Batstone *et al.* 2002

David Wagner
0981-203633-24
david.wagner@hs-ansbach.de

Biomasse Institut
www.biomasseinstitut.de
biomasse-institut@hswt.de
biomasse-institut@hs-ansbach.de

KIEff
Kompetenzzentrum
Industrielle Energieeffizienz

HOCHSCHULE ANSBACH

Material- und Energieflusssimulation in Aluminium-Druckgussbetrieben

Johannes Dettelbacher, Wolfgang Schlüter

Motivation

- Steigerung der Energieeffizienz ist ein wesentlicher Bestandteil der Energiewende
- Energie- und Kosteneinsparungspotenzial ist in energieintensiven Branchen, wie z. B. der Aluminium-Druckgussindustrie, besonders groß
- Entwicklung eines hybrides Simulationsmodell mit variabler Betriebsstruktur, welches beliebige Druckgussbetriebe und Steuerungsstrategien abbilden kann
- Simulationsmodell ermöglicht Untersuchung von Effizienzmaßnahmen ohne den realen Betrieb zu beeinflussen

Betriebsstruktur Druckgussbetriebe

- Analyse des grundsätzlichen Aufbaus verschiedener Betriebe zeigt die Unterschiede und Gemeinsamkeiten sowie ihre Auswirkungen auf den innerbetrieblichen Materialfluss (Abbildung 1)
- Kontinuierliche Prozessschritte (z. B. Schmelzen) und ereignisdiskrete Prozessschritte (z. B. Staplertransport) im Betrieb

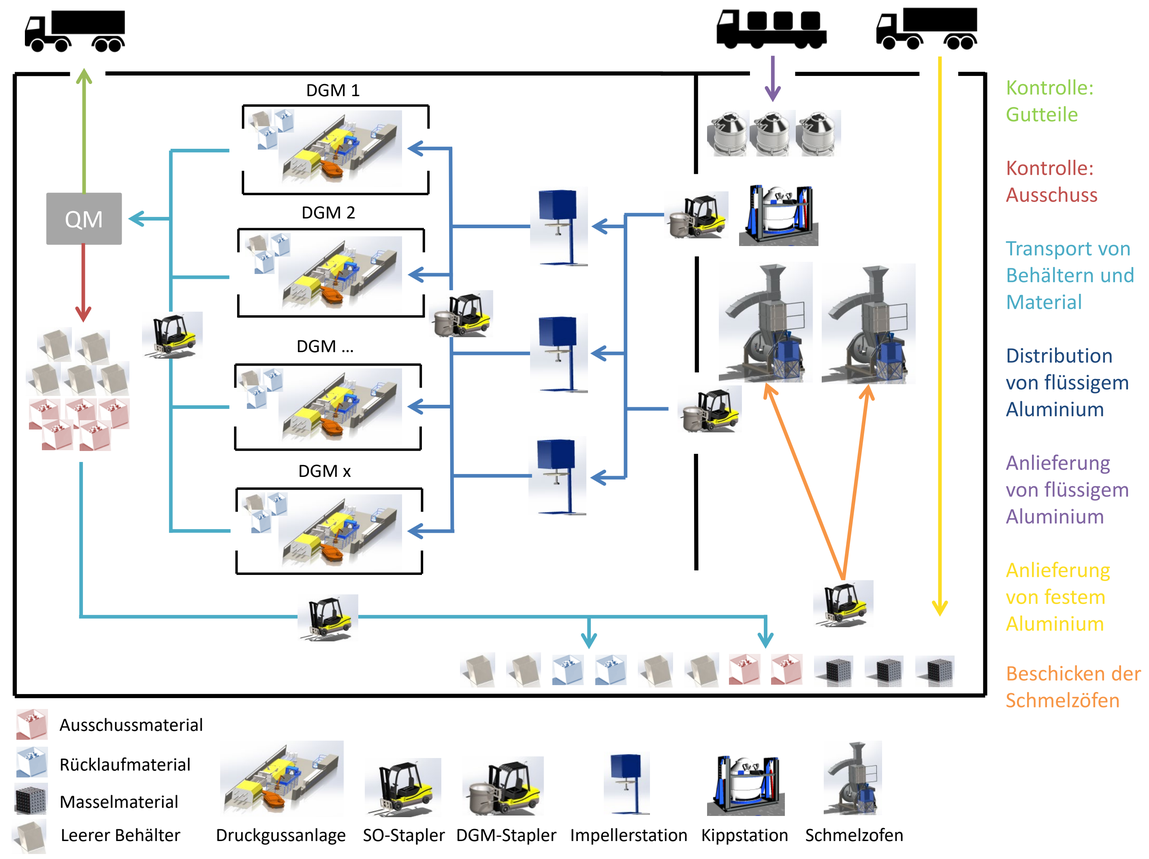


Abb. 1: Schematische Darstellung eines Aluminium-Druckgussbetriebes

Material- und Energieflusssimulation mit variablen Betriebsstrukturen

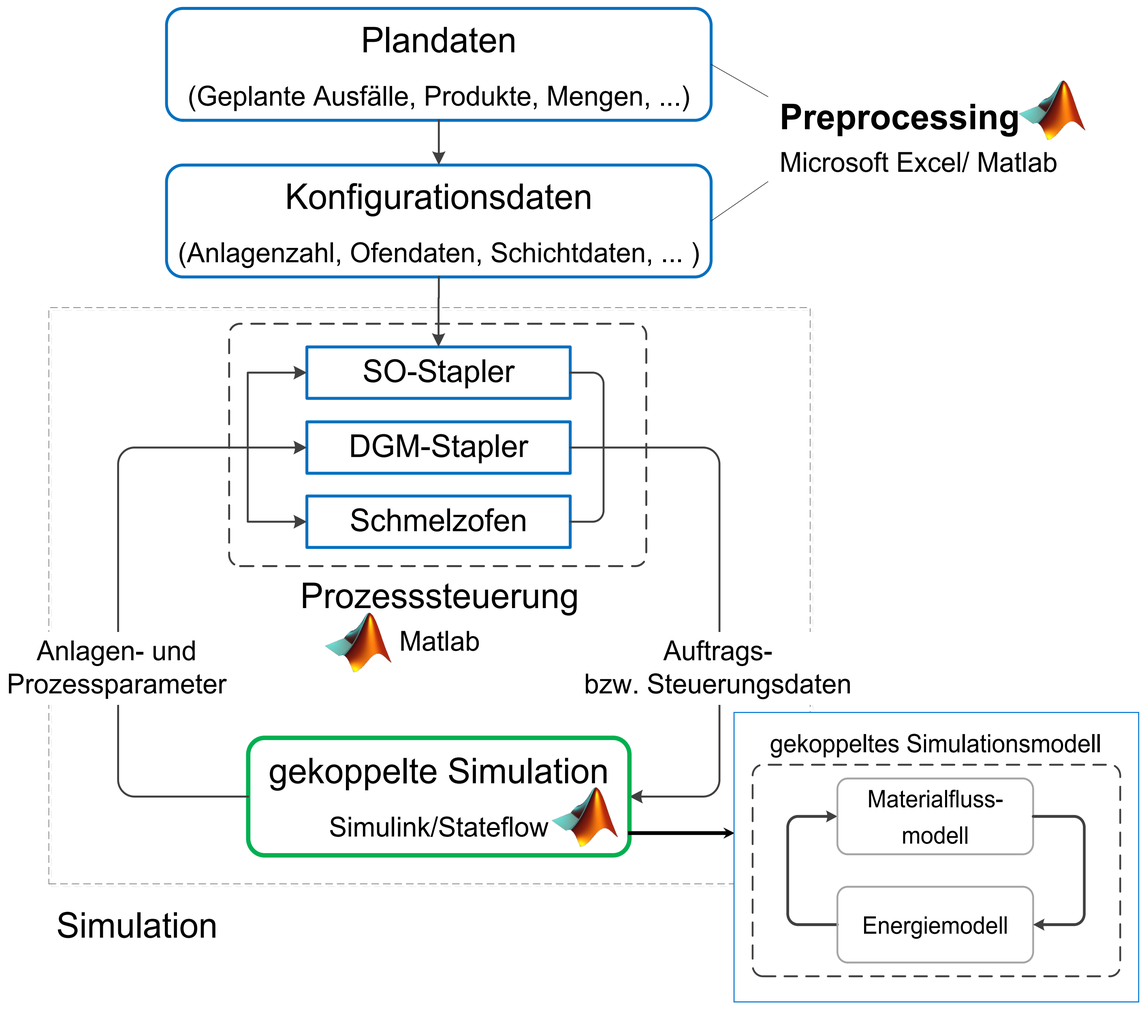


Abb. 2: Modellgenerierung und Simulation

- Simulationsmodell wird aus Plandaten und Konfigurationsmodell generiert (Untersuchung von verschiedenen Betriebsstrukturen möglich)
- Kernelemente des Simulationsmodells sind die Prozesssteuerung sowie das Material- und Energieflussmodell
- Material- und Energieflussmodell sind über ein Schnittstellenobjekt verbunden
- Simulationsmodell ist anhand von zwei realen Betrieben validiert

Ergebnisse und Online-Demonstrator

- Verbesserte Beschickungsstrategie der Schmelzöfen, Materialvorwärmung mit dem Abgasstrom und verbesserte Belieferung der Druckgussmaschinen wurden an unterschiedlichen Betrieben simulativ untersucht
- Spezifischer Energieverbrauch kann mit den Effizienzmaßnahmen um 25 % gesenkt und Anlageneffizienz (OEE) um 3 % gesteigert werden
- Aufbau eines Online-Demonstrator (Abbildung 3)

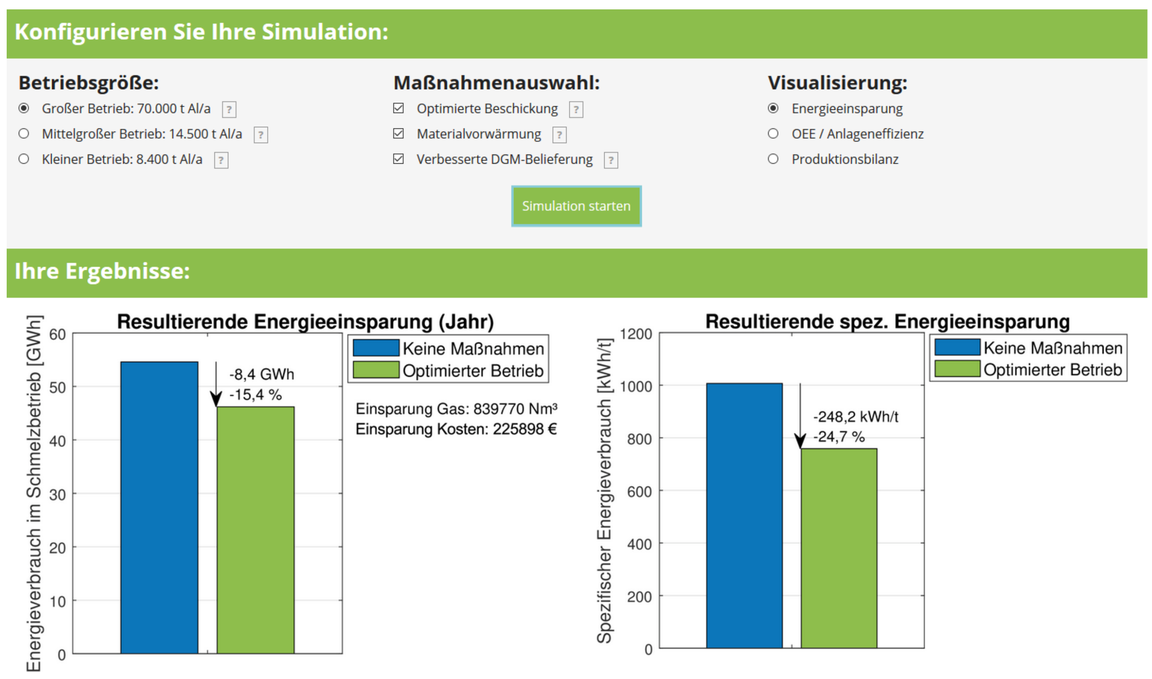


Abb. 3: Online-Demonstrator (www.hs-ansbach.de/emelt)

- Aufzeigen der Wirksamkeit der Energieeffizienzmaßnahmen für drei typische Betriebe verschiedener Größe
- Visualisierung von Energieeinsparung, Anlageneffizienz und Produktionsbilanz
- Demonstrator zeigt beispielhaft die Potentiale von Effizienzmaßnahmen; Für spezifischere Untersuchungen wird die hybride Simulation verwendet

Hochschule Ansbach
Residenzstraße 8
D-91522 Ansbach

Prof. Dr. Wolfgang Schlüter
Tel.: 0981 4877-317
wolfgang.schlueter@hs-ansbach.de

Johannes Dettelbacher, M. Sc.
Tel.: 0911 568549190
j.dettelbacher@hs-ansbach.de

Preview unavailable

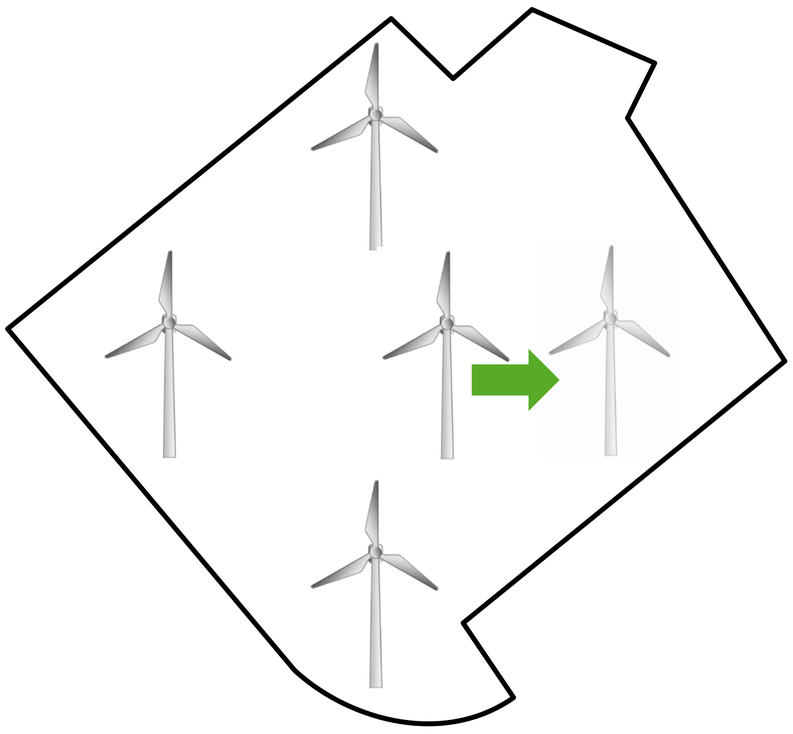
WiFO

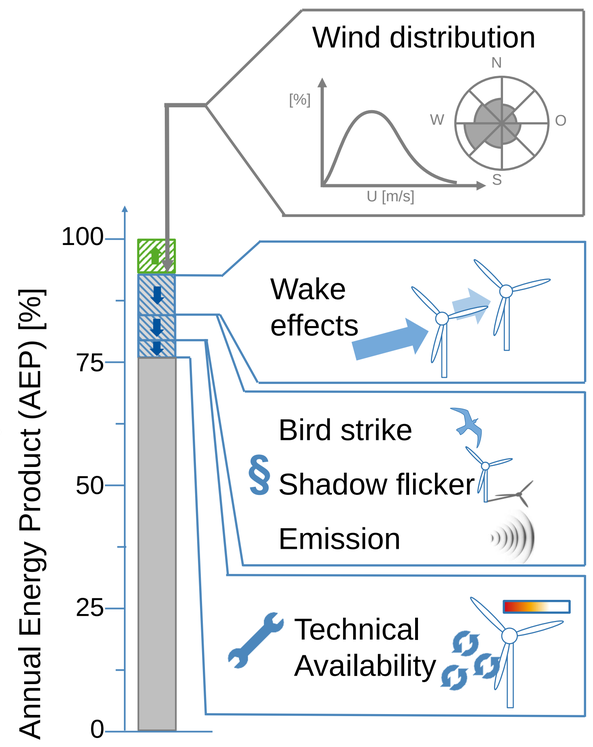
Multi-dimensional **Wi**nd **F**arm Layout **O**ptimization

Using limited area to its full potential

Targets:

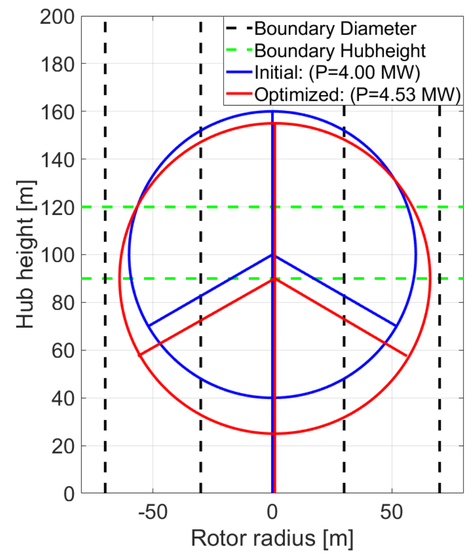
- Maximum annual energy product
- Minimal Cost of Energy
- Optimization of:
  - Wind turbine position
  - Wind turbine configuration
  - Internal grid
- Considering social aspects
  - Bird strike
  - Acoustic emission
  - Shadow flicker





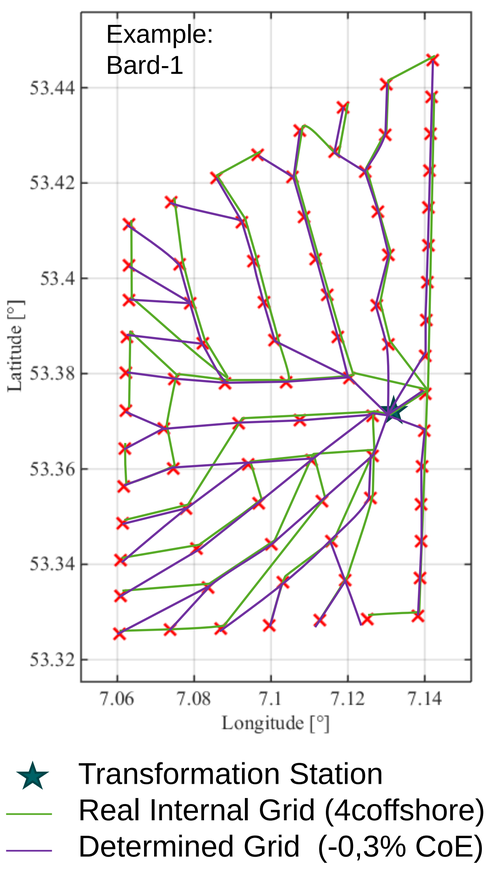
Wind turbine configuration

- Location based wind turbine design



Internal Grid consideration

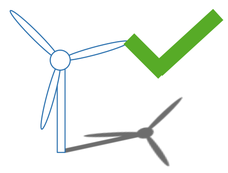
- Power loses due to Resistance
  - -1 till to -2% of the AEP
- Increase of CoE due to cable cost
  - +5% of CoE



AEP reducing matters

Shadow flicker prevention

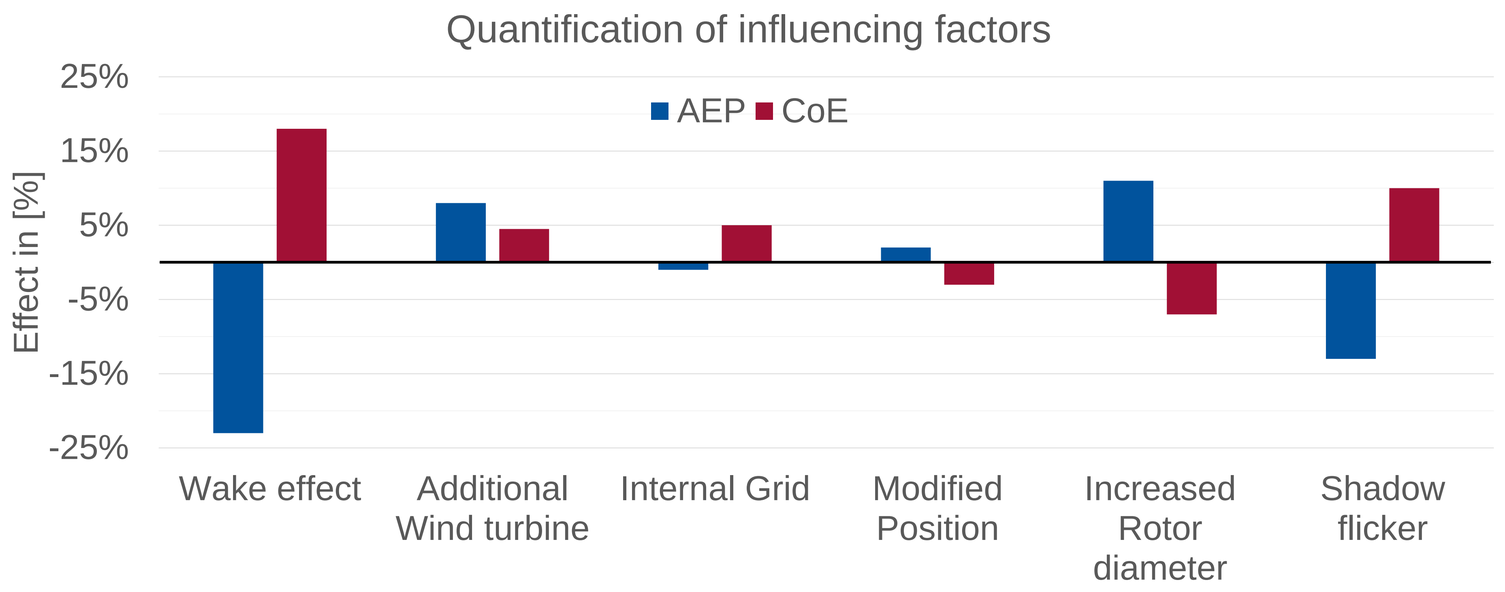
- Shut down of turbine when exceeding legal boundaries
  - 30 h/a or 30 min/day
- AEP reduction up to 10%



Shut down for bird strike prevention

- Shut down is probability based on weather condition etc.
- Depending on location
  - AEP reduction up to 20%

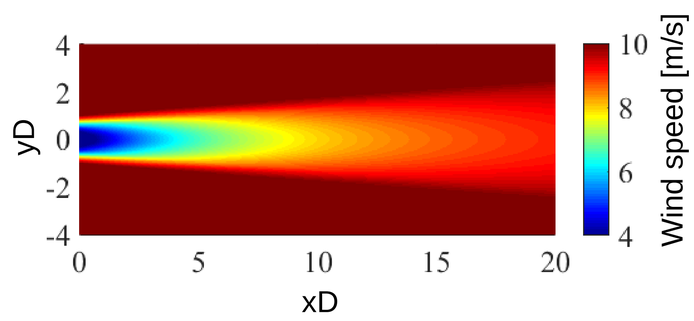




Methods

Flow simulation

- Modified Katic-Wake Modell
- Turbulence model by Frandsen



Encastered Optimization

- Genetic optimization to find wind farm layout
- Pattern search to find optimal wind turbine configuration
- Minimal spanning tree as internal grid determination

Björn Roscher, M. Sc.
Campus-Boulevard 61, 52074 Aachen, GERMANY
Tel: +49 241 80-95603
E-Mail: Bjoern.roscher@cwd.rwth-aachen.de
https://www.cwd.rwth-aachen.de

02.19 BR

CWD Chair for Wind Power Drives | RWTH AACHEN UNIVERSITY

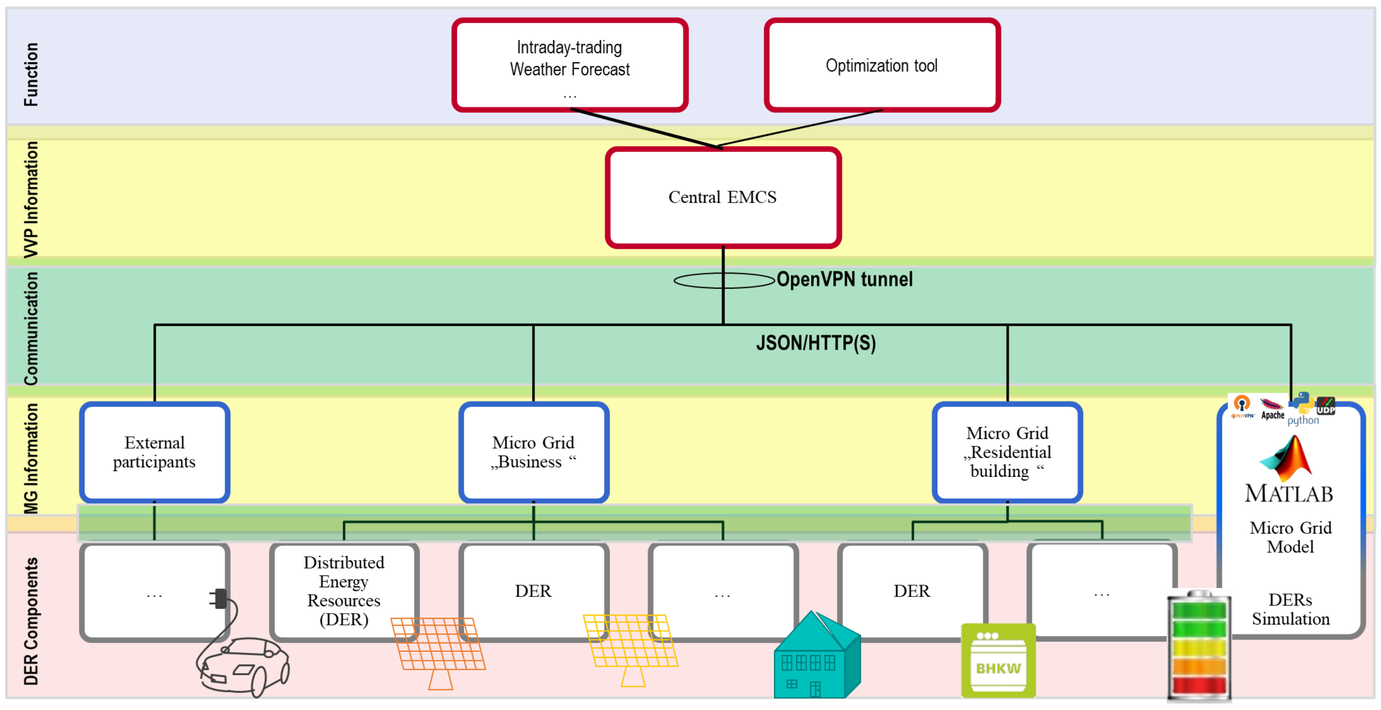
Smart Grid modeling in MATLAB-Simulink®

M.Sc. Petr Tugarinov, Dr. Bernhard Nold, Prof. Dr.-Ing. Frank Truckenmüller
Reutlingen University, Faculty of Technology, 72762 Reutlingen, Alteburgstraße 150

Abstract

Virtual Power Plant (VPP) allows to the increasing number of Distributed Energy Resources (DERs) participate at the energy market as a larger entity. DERs can be controlled by an energy management and control system (EMCS), that provides flexibility for matching energy demand and supply.

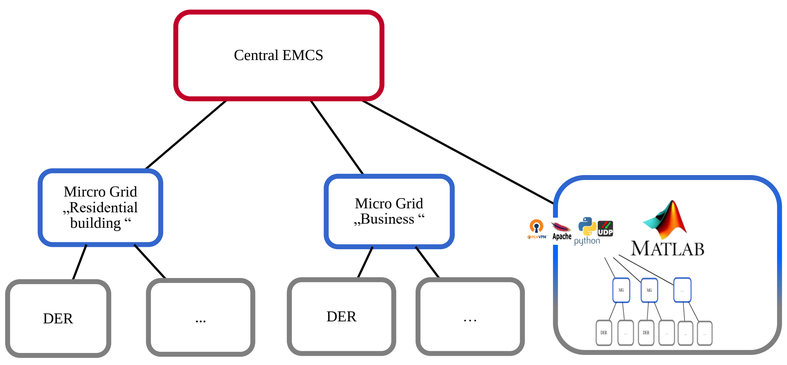
Virtual Power Plant Neckar-Alb



- The Demonstration Project Virtual Power Plant Neckar-Alb (VKNA) is constructing a VPP demonstration site at the Reutlingen University campus. Project aims at building a demonstration platform for study different approaches for operation, optimization and control DERs.
- A number of devices, such as Combined Heat and Power (CHP), Photovoltaics, E-Car Charging station, Adsorption chiller, Energy storage, were purchased and installed at the Reutlingen University campus. Off-campus DERs are owned by external partners of the project and located at the premises of their respective owners.
- Building a Hardware-in-the-Loop Systems in MATLAB provides flexibility of Smart Grid energy management and broadens the horizons for studying and developing new optimization solutions.

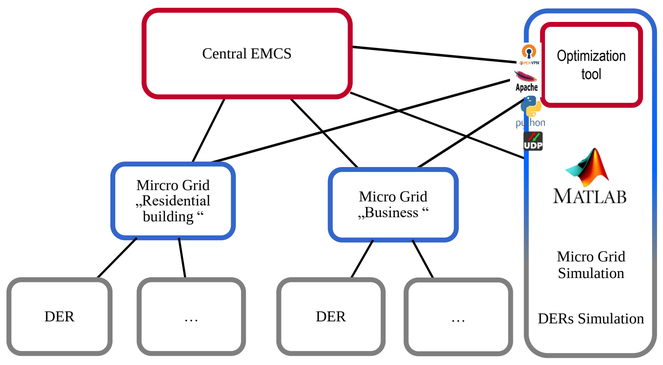
Study cases

Distributed Optimization



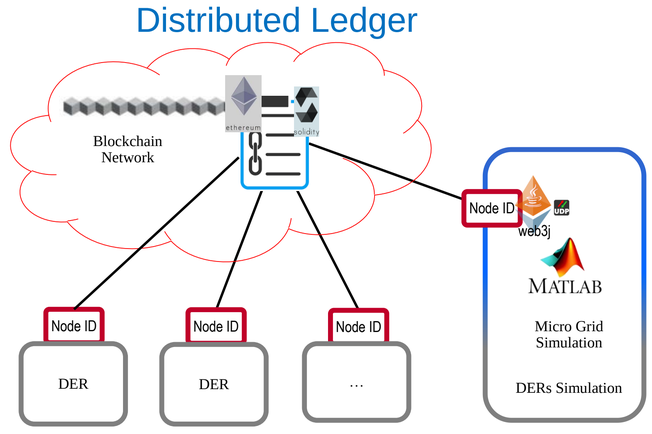
Both the MG controllers and MATLAB can create local working schedule. The optimization and scheduling of DERs is performed at the local DER controllers. The central EMS provides monitoring and visualization of all DERs including MATLAB model.

Parallel Optimization



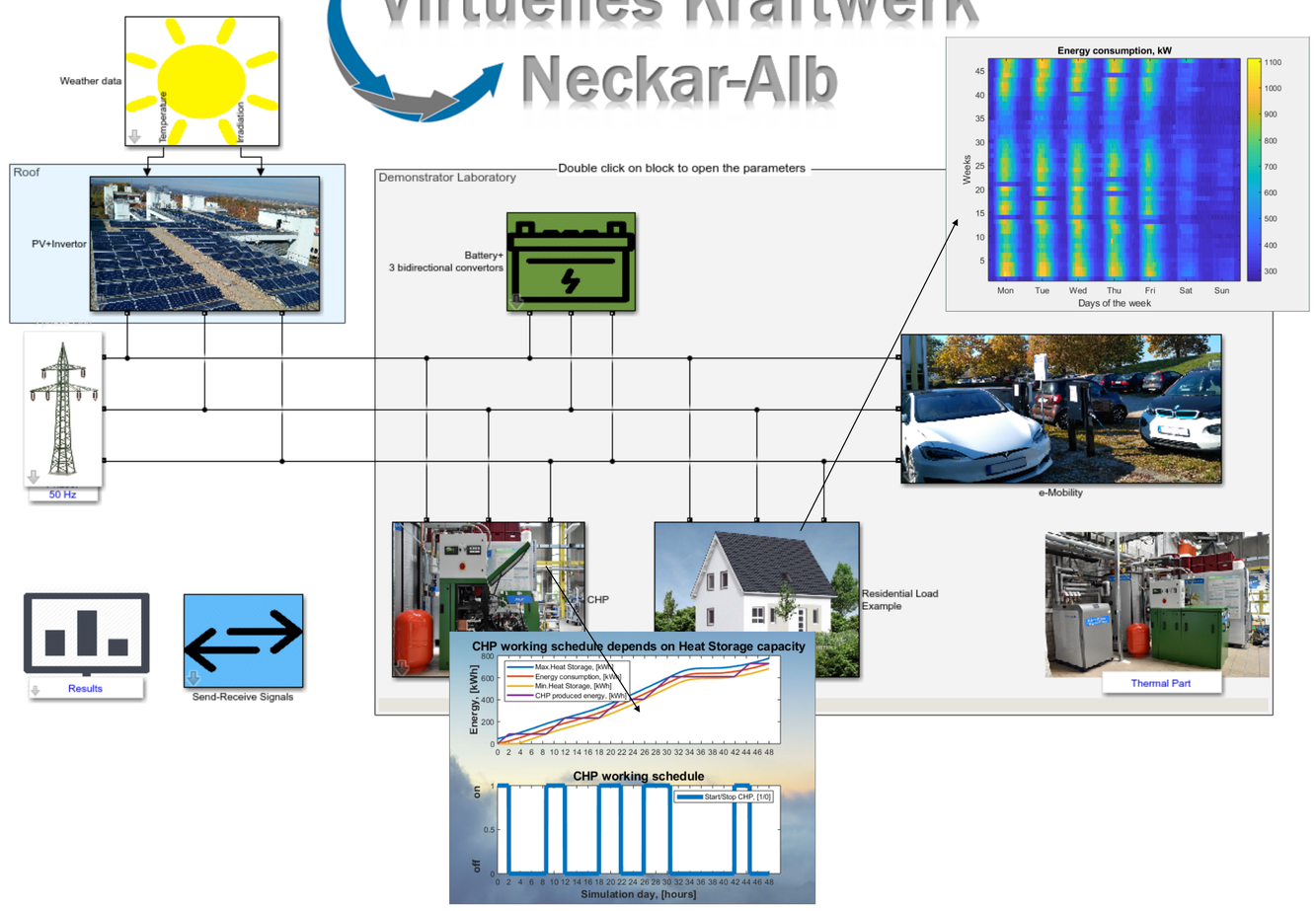
The parallel demonstration of different controller implementations for different sets of DERs. This way, the EMCS and the MATLAB server can both independently manage and optimize a subset of the DERs.
By connecting MATLAB Optimization Tool it becomes possible to study already existing and develop new optimization solutions.

Distributed Ledger



Distributed Ledger demonstration can be used for designing a business case and implementing a concept for using a smart contract within a private Blockchain Network to represent the trade of energy between multiple actors.
MATLAB simulation allows to create plenty of study cases with different electricity prices and DERs capacities.

Smart Grid Model



- MATLAB software is installed on a Virtual Machine on an University server.
- After every server restart it will automatically establish connection to the EMSC and autorun Simulation in autonomous mode.
- Model includes electrical and heating parts with representatives of several DERs installed at the University laboratory, such as CHP, Photovoltaics, Heat Storage, etc.
- Weather data could be taken from the Weather Station, which is installed at the campus, from EMCS or from open Weather API.
- Different profiles of the Electrical Loads and Heat Demand can be used for simulation.
- The Model receives control signal from the EMCS and sends it to the units‘ working schedule, or it can generate its own schedule for each DER.
- Intermediate results are packaged and sent via UDP to a Python program, which receives the signal and records it in JSON format, which is required for the data exchange with management system.
- Apache and OpenVPN software respectively provide webserver and secure connection to the EMCS.

Contact: **Petr Tugarinov**
+49 (0)7121 271-7137
petr.tugarinov@reutlingen-university.de

Hochschule Reutlingen
Alteburgstrasse 150
72762 Reutlingen