



Technische
Universität
Braunschweig

Institut für
Flugführung



Übung – Anwendungsorientierte Programmierung für Ingenieure Einführung

Prof. Dr.-Ing. Peter Hecker, Dipl.-Ing. Paul Frost, Dipl.-Ing. Tobias Rad,
05. April 2016

Teil I

Institutsvorstellung



Institut

Institut für Flugführung
Hermann-Blenk-Str. 27
38108 Braunschweig
Prof. Dr.-Ing. Peter Hecker

**Institut für
Flugführung**



Technische
Universität
Braunschweig

05. April 2016 | Prof. Dr.-Ing. Peter Hecker, Dipl.-Ing. Paul Frost, Dipl.-Ing. Tobias Rad | Seite 3
Übung – Anwendungsorientierte Programmierung für Ingenieure

Institut für
Flugführung



Flugführung

*Verfahren und technische Mittel zur Unterstützung
des Menschen beim Führen von Flugzeugen*



Abbildung 1: Heinrich
Koppe, 1925



Abbildung 2: Bereiche der Flugführung

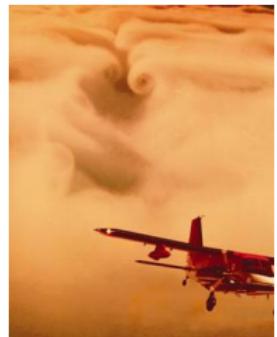
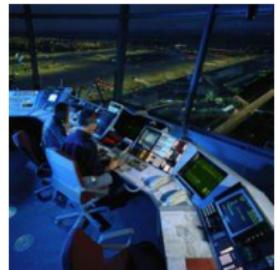


Motivation

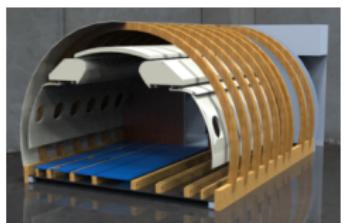


Forschungsschwerpunkte des IFF

- Integriertes Air Traffic Management
- Präzisionsortung und -Navigation
- Modellierung und Simulation dynamischer Systeme
- Flugmesstechnik
- Wirbelschleppen
- Mensch-Maschine-Schnittstelle
- Fluggravimetrie



- Forschungsflugzeuge
 - Cessna 172
 - Dornier 128-6
- Forschungsfahrzeug
 - VW Passat
- ATM-Simulatoren
 - Airbus A320 Cockpit
 - Airbus A320 Kabine
 - Diamond DA42
 - Piloten-
/Lotsenarbeitsstationen



- Testsysteme Satellitennavigation
 - Galileo Umgebung ÄviationGate"
 - Galileo Labor (Simulator)
- Präzisionsnavigationssysteme
 - Inertialmesseinheiten
 - Gekoppelte Systeme
- Unmanned aerial vehicle (UAV)
 - Quadrocopter
 - Rover
- Hubschrauberschleppsonde
 - HeliPod



Teil II

Anwendungsorientierte Programmierung für Ingenieure (API)



Betreuer

Paul Frost

p.frost@tu-braunschweig.de

0531/391-9826

Tobias Rad

t.rad@tu-braunschweig.de

0531/391-9844

- Sprechstunden: nach Vereinbarung
- Fragen per E-Mail
- Austausch über StudIP-Forum



Ingenieursarbeit und -ausbildung

- Analyse und Entwicklung technischer Systeme
- Problemlösung mit Hilfe von Software
 - Simulation
 - Optimierung
 - Datenanalyse
 - Entwurf von Reglern
- Software als Ergebnis der Entwicklung
 - Schnittstellen zur Aufzeichnung und Überwachung
 - Implementierte Regler
 - Ausgabe von verarbeiteten Daten



Anwendungsorientiert

Berufspraxis

Betrachtung der Arbeitsabläufe in der Praxis

- Rechtliche Bedingungen
- Projekt- und Risikomanagement
- Zusammenarbeit mit Softwareentwicklern
- Entwicklungswerzeuge

Programm

Verwendung einer Computeranwendung zum Lösen von Ingenieurproblemen

- MATLAB/SIMULINK
- Entwicklung für spezialisierte Hardware
- Datenauswertung



Inhalte

- Umgang mit Projekten, die Softwareentwicklung erfordern
- Kennenlernen von Werkzeugen und Verfahren in der Softwareentwicklung
- Kommunikation und Informationsaustausch mit Informatikern
- Entwicklung von Software für spezielle Hardware oder besondere Umgebungen
- Bearbeiten typischer Ingenieuraufgaben mit MATLAB und SIMULINK
- Einsatz von Werkzeugen der Softwareentwicklung im Studium



MATLAB

Vorstellung MATLAB

- Installation
- Anwendungsbereiche
- Vor- und Nachteile

Grundlagen

- Arbeitsumgebung
- Variablen
- Grundlegende Funktionen
- Plots

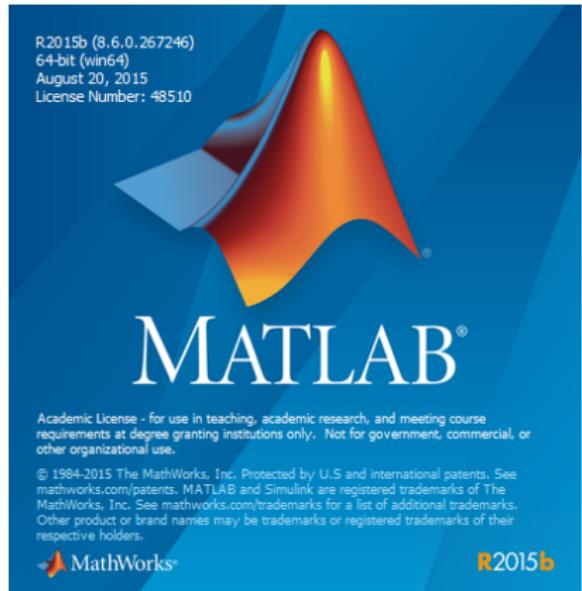


Abbildung 3: Über MATLAB



MATLAB

Skriptentwicklung (m-Files)

- Funktionsgenerierung
- Debugging
- Profiling
- Datei-I/O

GUI Entwicklung

- Callback-Funktionen
- Einbindung in m-Files

The screenshot shows the MATLAB IDE with the 'f.m' file open in the editor. The code in the file is:

```
1 %>>> function f()
2 %>>> % set up the interval vector, initial conditions and options
3 %>>> tspan=[0 10];
4 %>>> ts = tspan(1)/0.01:tspan(2);
5 %>>>
6 %>>> x0=[0 0];
7 %>>> options = odeset('Stats','on','OutputFcn',@odeplot);
8 %>>>
9 %>>> % Run ode45
10 %>>> [T,X] = ode45(@g,tspan,x0);
11 %>>>
12 %>>> plot(T,X);
13 %>>> hold on;
14 %>>> plot(ts,h(tts),'x');
15 %>>> hold off;
16 %>>>
17 %>>> ylabel('m / m') % m / (m/s) * s
18 %>>> legend('x / m','y / (m/s)*s(w(s))');
19 %>>>
20 %>>> end
21 %>>> function da_dmg(t,y)
22 %>>> m = 1;
23 %>>> c = 5;
24 %>>> d = 0;
```

Abbildung 4: MATLAB m-File Editor



SIMULINK

Generell

- Grundfunktionen
- Solver Eigenschaften
- Vorstellung von Toolbox-Funktionen

Projektbezogen

- Generierung einer Testumgebung
- Verarbeitung von Sensordaten
- Erstellung einer Netzwerkschnittstelle

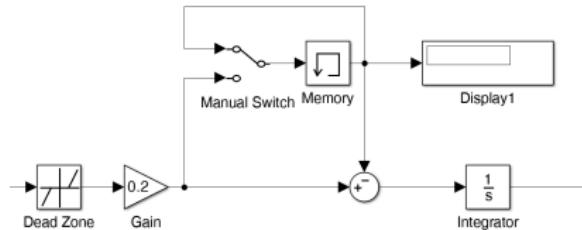


Abbildung 5: SIMULINK Beispielprojekt



Plattformübergreifende Entwicklung

- Programmieren für Android unter Windows
 - Grundsätzlicher Aufbau
 - Zugriff auf Sensoren
 - Speichern und Versenden von Daten
- Verwendung des MATLAB Hardware Support Package
- Schnittstellenanbindung
- Einbindung in das Projekt

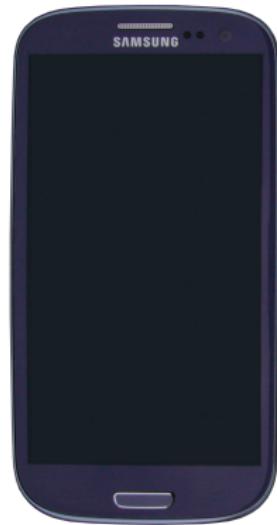


Abbildung 6: Samsung Galaxy S3

Illythr, CC BY-SA 3.0

http://commons.wikimedia.org/wiki/File:Samsung_Galaxy_S_III_Pebble_Blue.PNG



Lernziel

- Verständnis für die Aufgaben und Probleme bei der Softwareentwicklung
- Erkennen der Bedeutung des Entwicklungsprozess abseits der Programmierung
- Einbindung der Software in den Problemlösungsprozess des Ingenieurs



Agenda

- 05. April Einführungsveranstaltung
- 12. April MATLAB Vektoren und Matrizen
- 19. April Daten und Grafiken
- 26. April Skripte und Funktionen 1
- 03. Mai Skripte und Funktionen 2
- 10. Mai Skripte und Funktionen 3
- 17. Mai **Exkursionswoche**
- 24. Mai GUI-Entwicklung
- 31. Mai Debugging, Testing und Profiling
- 07. Juni Simulink 1
- 14. Juni Simulink 2
- 21. Juni Simulink 3
- 28. Juni Plattformübergreifende Softwareentwicklung 1
- 05. Juli Plattformübergreifende Softwareentwicklung 2
- 12. Juli **Klausurvorbereitung**



Ablauf einer Übung

- Wiederholung der vorhergehenden Übung
- Vermittlung der Grundlagen mit Folien
- Einführung in die Aufgabe
- Individuelle Bearbeitung der gestellten Aufgaben in Gruppen
 - Bearbeitung am Laptop
 - oder
 - Schriftliche Bearbeitung
- Besprechung und Ausarbeitung der Lösung

→ weniger Frontalunterricht



Veranstaltungsort und Material

Übung

- 14 Termine
- Zwei Semesterwochenstunden
- Dienstag 16:45 – 18:15
- Pk 15.1
- HS 65.2 PC-Pool (Live-Übertragung der Übung)

Stud.IP

- Präsentationsfolien in Stud.IP (<https://studip.tu-bs.de>)
- Verfügbar spätestens am Tag der Vorlesung
- Kein Passwortschutz
- Forum zur Diskussion über Übungsinhalte

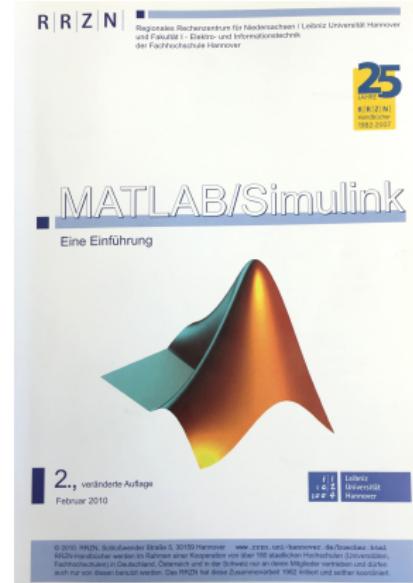


Skript



Rüdiger Kutzner und Sönke Schoof:
MATLAB/Simulink – Eine Einführung,
Hrsg: RRZN / Leibniz Universität
Hannover (2014)

Bestellbar über eine Umfrage in StudIP!



Prüfung

- Modulprüfung Computergestützte Methoden für Ingenieure
- Termin: **19. Juli 2016**
- Anwendungsorientierte Programmierung für Ingenieure: 60 Minuten
- Es gibt keinen Fragenkatalog oder alte Klausuren
- Alle Inhalte aus der Übung sind relevant für die Klausur
- Wissensüberprüfung mit Hilfe von Stud.IP und den Wiederholungen der Übung
- Programmierung nicht C/C++, sondern MATLAB/Simulink



Teil III

Matlab



Programmpaket MATLAB/SIMULINK

MATLAB

- Software zur Lösung mathematischer Probleme
- Ursprünglich für numerische Berechnungen von Matrizen
MATLAB = Matrix Laboratory
- Entwickelt in den 1970er-Jahren in der UNIVERSITY OF NEW MEXICO
- Heute: Produkt der Firma THE MATHWORKS (1984)

SIMULINK

- MATLAB-Erweiterung
- Graphische Modellierung von Systemen durch Blöcke
- Bibliothek mit vorgefertigten Blöcken
- Numerische Simulation



Typische Anwendungsbereiche

MATLAB

- Entwicklung von Algorithmen
- Datenanalyse und Visualisierung
- Entwurf grafischer Oberflächen
- Rapid Application Development

SIMULINK

- Regelungstechnik
- Signalverarbeitung



Lizenzierung

Rahmenlizenz TU Braunschweig

- Für Angehörige der TU Braunschweig
 - Installations-DVD beim GITZ (10 € Schutzgebühr)
 - Download über
[https://campus-software.tu-braunschweig.de/index.php?
dir=MATLAB](https://campus-software.tu-braunschweig.de/index.php?dir=MATLAB)
- Netzwerk Lizenz – Nutzung nur mit IP-Adresse der TU Braunschweig
 - Eduroam-Netzwerk
 - VPN der TU-Braunschweig
- Einsatz nur für Forschung und Lehre zulässig
- Anleitungen auf GITZ Seiten der TU Homepage:
<https://doku.rz.tu-bs.de/doku.php>



MATLAB Toolboxen

- Erweiterung der Funktionalität
- Vertrieb zum Teil durch THE MATHWORKS, z. B.:
 - Optimization Toolbox
 - Statistics Toolbox
 - Aerospace Toolbox
 - Fuzzy Logic
 - Econometrics Toolbox
 - ...
- Verfügbare Lizenzen können über folgenden Link eingesehen werden:
<http://gitz-dl.tu-bs.de/software/matlab.pl>



Entwicklungsumgebung

- Workspace
- Command Window
- Current Folder
- Status:
 - Ready
 - Busy

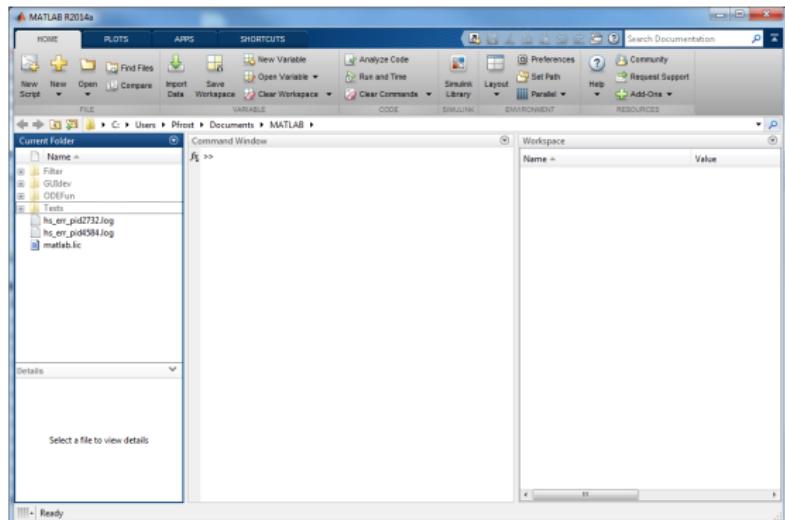


Abbildung 7: MATLAB Entwicklungsumgebung



Command Window

- Befehlszeile zur Bedienung von MATLAB
- Ergänzen von Befehlen durch Tabulator
- Wiederholung vorheriger Befehle durch Pfeiltasten
- Anzeige von Kurzinformation zu einer Funktion mit:
help NameDerFunktion



The screenshot shows the MATLAB Command Window with the following text:

```
Command Window
>> help help
help - Help for functions in Command Window

This MATLAB function lists all primary help topics in the Command Window.

help
help name

Reference page for help

See also class, dbtype, doc, lookfor, more, path, what, which, whos

Other functions named help
    simulink/help, dsp/help, distcomp/help, stateflow/help

f1 >> |
```

Abbildung 8: MATLAB Command Window



Workspace

- Anzeige aktiver Variablen
- Anlegen und Bearbeiten von Variablen
- Statistische Daten von Variablen werden angezeigt
- Variablen können auch in SIMULINK verwendet werden

Name	Value
f	1x513 double
Fs	1000
L	1000
NFFT	1024
t	1x1000 double
T	1.0000e-03
x	1x1000 double
y	1x1000 double
Y	1x1024 complex double

Abbildung 9: MATLAB Workspace



Produkthilfe

- Befehl:
doc dot
- Dokumentation für
 - Syntax
 - Eingabeparameter
 - Ausgabewerte
- Suchfunktion
- Übersicht
- Beispiele

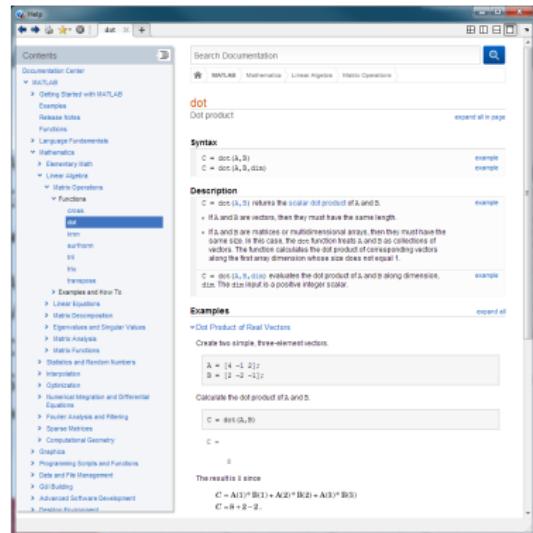


Abbildung 10: MATLAB Dokumentation



Gibt es noch Fragen?

