Einführung in CAE-Systeme

Dipl.-Ing. Mark Müller

Lehrstuhl für Regelungstechnik und Signaltheorie

Inhalt

1. Motivation für CAE-Werkzeuge
2. Modellierung technischer Prozesse
3. Übersicht über CAE-Simulationssysteme
   - Kommerzielle Programme
   - Freeware
4. Funktionsinhalte von Simulationssystemen
5. Beispiel MATRIXx

1. Motivation für CAE-Werkzeuge

Klassische Entwurfsmethoden der Regelungstechnik

- lineare Entwurfsmethoden (Frequenzkennlinie,WOK)
- nichtlinearer Entwurf (z.B. Phasenebene)
- Optimierungsverfahren
- Zustandsregler

Neuere Entwurfsmethoden

- Fuzzy-Control
- Expertenregler
- Hybride Regelstrukturen

Modellierung technischer Prozesse

Generierung von Source-Code

2. Modellierung technischer Systeme

Problemstellung:

- Keine geschlossene mathematische Lösung angebbar
- Test an realem Prozess nicht möglich

Ursprüngliche Vorgehensweise (ohne CAE):

1. Strukturbild & Systemgleichungen
2. Diskretisierung und Verwendung von numerischen Routinen zur Integration
3. Implementierung mit Programmiersprachen (Basic, C/C++, Pascal, FORTRAN etc.)
4. Simulation

Nachteil der Implementierung in Quellcode:

- Umsetzung in diskretes Simulationsmodell
- Programmierkenntnisse
- Abkehr von der eigentlichen Aufgabe, dem Reglerentwurf
- Individuelle Simulationsprogramme
  intransparent, fehlende Kompatibilität
- Fehleranfällig
- Zeitraubend (Fehlersuche)
- Komplexe Simulationsprogramme

Vorgehensweise mit CAE-Werkzeugen:

1. Strukturbild & Systemgleichungen meist bereits in graphischer Form angebbar
2. Entfällt !
3. Entfällt !
4. Simulation
5. Kritisches Bewerten der Simulationsergebnisse

Vorteile:

- Keine Programmierkenntnisse
- Konzentration auf Modellbildung und Simulation
- Weniger fehleranfällig
- Transparent
- Zeit- und kosteneffizient
- Leicht dokumentierbar

Nachteile:

- Kauf eines CAE-Simulationssystems (Hard- und Software)
- Numerische Rechenfehler Fehlinterpretationen, oberflächliche Analyse (wahlloses Simulieren)
- Fehlinterpretation aufgrund ungenügender Kenntnis der verwendeten Rechenverfahren

Übersicht über Simulationssysteme – kommerziell:

MATLAB : Modellbildung, Simulation, Reglerentwurf, Matrixalgebra

MATRIXX: Modellbildung, Simulation, Reglerentwurf, Matrixalgebra

MATHEMATICA: symbolische und numerische Mathematik

MAPLE: symbolische Mathematik

MSC.ADAMS: Modellbildung, Simulation, Reglerentwurf, Matrixalgebra, FEM-Untersuchung

MODELICA, DYMOLA, DYNASIM: Chemische Modelle, elektrische Schaltkreise, Petrinetze, Zustandsdiagramme, thermodynamische Systeme, Hybridesysteme, ...

Übersicht über Simulationssysteme – kommerziell:

LabView: Messwerterfassung und Verarbeitung

PSPICE: Elektrische Schaltkreise

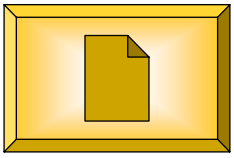
gPROMS: chemische Prozessmodelle

DORA-PC: Regelungstechnik, Entwurf, Analyse, Simulation http://esr.e-technik.uni-dortmund.de/Software_DfW.htm

MADONNA: Differentialgleichungen, graph. Ausgaben

http://www.berkeleymadonna.com/

....



Software Guide
mit Preisliste

3. CAE-Simulationssysteme

Übersicht über Simulationssysteme – Freeware:

SCILAB: Matrixalgebra, Reglerentwurf, Simulation, FEM, ... (Win/Linux/Mac)
http://scilabsoft.inria.fr/

ASCEND: Chemical engineering
http://www.cs.cmu.edu/~ascend/

ITI SimulationX: Simulation von Fahrzeugen und Antriebssträngen (kostenlose Studi-Ver.)
http://www.iti.de

4. Funktionsinhalte

Grundsätzlich:

- mathematische Operationen
  - insbesondere Matrix-Operationen
- Systemanalyse
  - Zustandsdarstellung dynamischer Systeme
- Reglersynthese
- Simulation

Weiterhin:

- Fuzzy-Control-Systeme
- System-Identifikation
- Modellvereinfachung
- Adaptive Regelungen
- Neuronale Netze

...

4. Funktionsinhalte

Weiterhin häufig:

- Blockorientierte Modellierung möglich
- Modellbibliotheken
- Anbindung von Hardware
- Echtzeitsimulation
- Software- und Hardware-in-the-Loop (SiL und HiL) Simulation möglich
- Autocodeerzeugung
- Steuergerätetest
- Rapid-Prototyping

...

5. Beispiel MATRIXx

Modularer Aufbau:

- Core Xmath: mathematisches Analysepaket mit
  - Komandozeilen-Interpreter,
  - Matrizendarstellung und –berechnung,
  - Komplexe Zahlen,
  - Graphischer Datenanzeige,
  - Objektorientierter Skriptsprache,
  - Mathematische Grundfunktionen (sin, cos, ...)
  - Erweiterbar mit Funktions-Bibliotheken
  - Module wie: Control Design,
    Robust Control,
    Signal Analysis, ...

Modularer Aufbau:

Xmath – Control Design Modul:

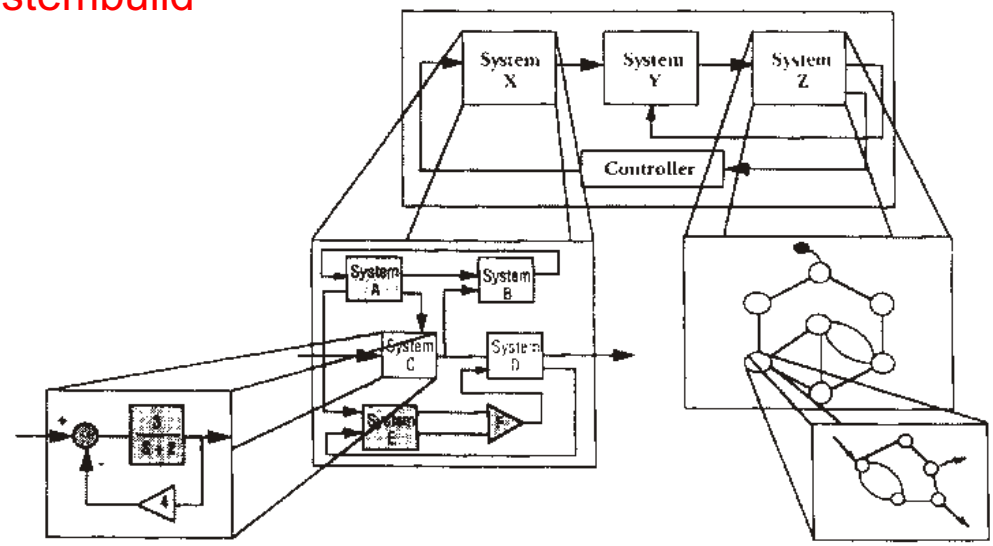
- Differentialgleichungen
- Übertragungsfunktionen
- Pol/Nullstellen-Darstellung
- Zustandsdarstellung
- Reglerentwurf nach
  - Frequenzkennlinien
  - Wurzelortskurve
  - Zustandsregler

Modularer Aufbau:

- Systembuild:
  - Graphisches Userinterface,
  - Hierarchische Modellstruktur (Superblocks)
  - Katalog-Browser,
  - Online-Hilfe,
  - Viele Erweiterungsmodule für Neuronale Netze, Fuzzy-Control, Zustandsautomaten, ...

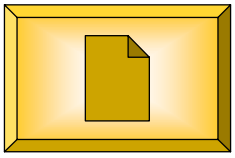
Modularer Aufbau:

- Systembuild



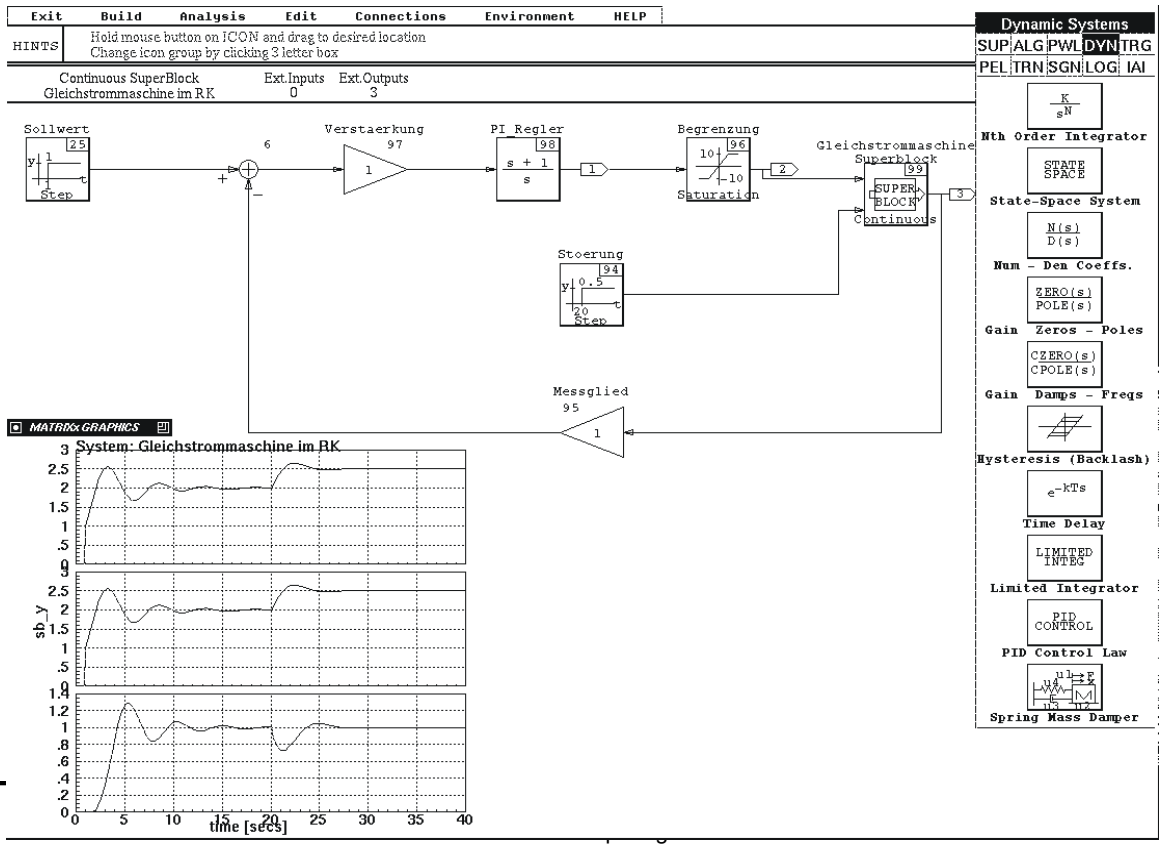
Modularer Aufbau:

- AutoCode: Echtzeitcode Generierung, für verschiedene Prozessoren optimiert
  - C und Ada
  - Hardware-in-the-Loop
- Dokument it: automatische Erstellung von Dokumentationen aus den Modellen

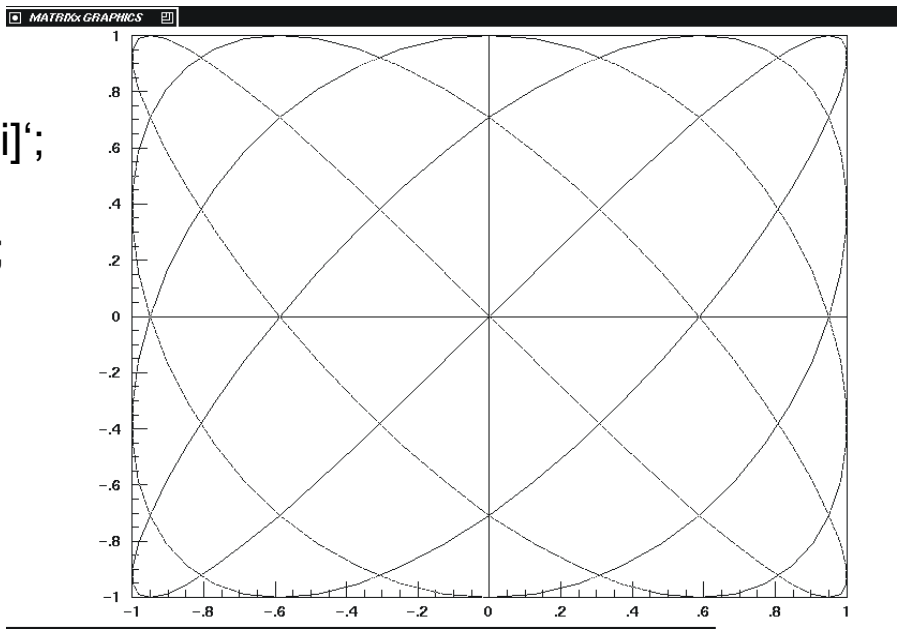


MatrixX mit Toolboxen
Preisliste

5. Beispiel MATRIXx



5. Beispiel MATRIXx

Beispiel Lissajous-Figuren

```
t = [0:pi/200:pi]';
x1 = sin(8*t);
x2 = sin(10*t);
plot(x1,x2)
```