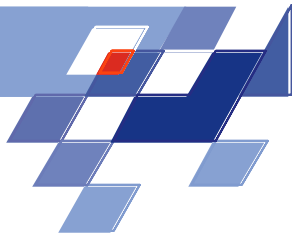
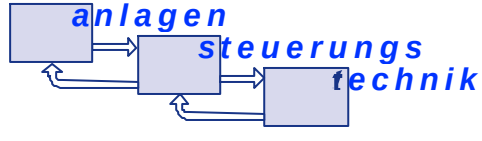


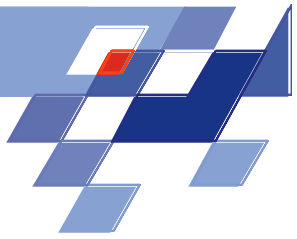
Forschungsband Modellbildung und Simulation,
Kolloquium vom 2002-06-10

Objektorientierte Modellierung und Simulation kontinuierlich-diskreter Systeme

Manuel Remelhe

- Kooperation im Rahmen des DFG-SPP KONDISK
  (Analyse und Synthese kontinuierlich-diskreter technischer Systeme):
  - LS AST, Dortmund: S. Engell, A. Deparade, M. Remelhe
  - DLR, Oberpfaffenhofen: M. Otter, P. Mosterman





Übersicht

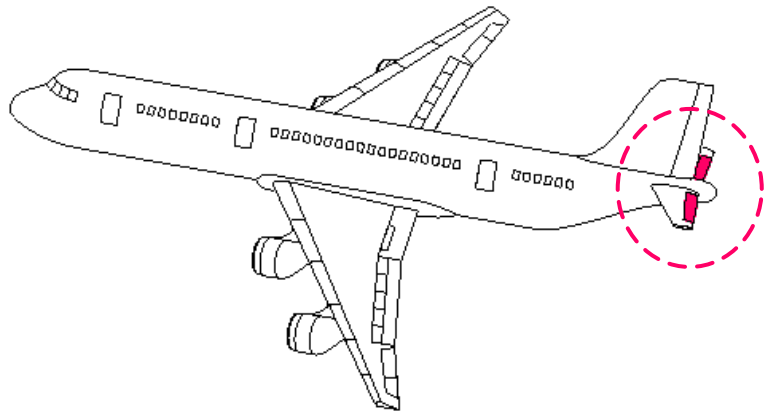
- Beispiel: Höhenrudersystem
- Anforderungen & Stand der Technik
- Problemlösung Teil 1:
  - objektorientierte Modellierung, Modelica
- Problemlösung Teil 2:
  - Integration ereignis-diskreter Modelle
- Zusammenfassung und Ausblick

Höhenrudersystem

Sicherheitskritisches System

➔ Redundante Subsysteme

➔ Redundanzmanagement



Validierung erfordert:

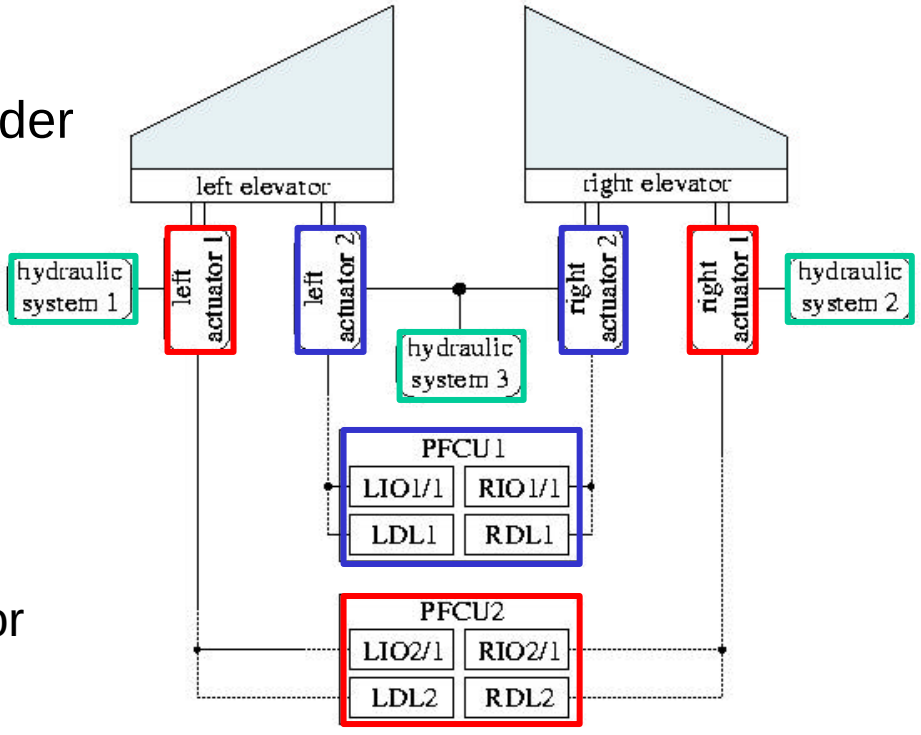
- Flugzeugmodell (Aerodynamik, Gravitation, etc.)
- Detailliertes Modell des Höhenrudersystems:
  - Redundanzmanagement, Regelungen
  - Hydraulische Aktuatoren, ...

Systemkomponenten

- 3 Hydraulik-Systeme
- 2 Aktuatoren pro Höhenruder
- 2 Steuergeräte
- 2 Regelungsmodi
  - IO: closed loop control
  - DL: open loop control

Redundanzmanagement:

- 1 Statechart pro Regler
  - aktiviert Regler & Aktuator
- Aufgabe:
  - 1 aktiver Aktuator pro Ruder
  - 1 aktiver Regler pro Aktuator

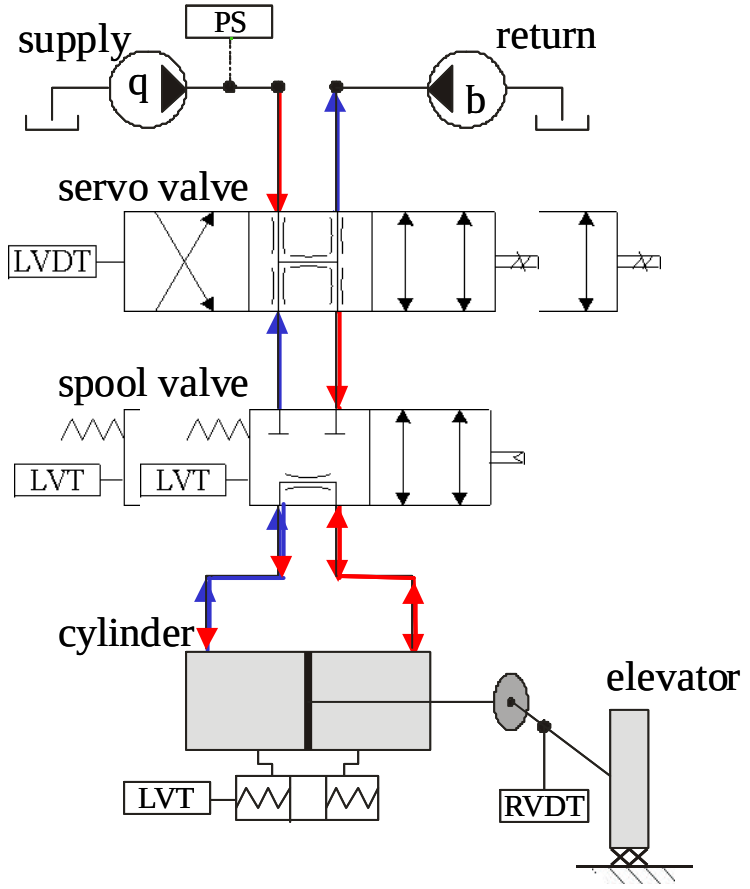


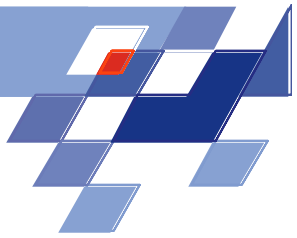
➔ **8 kommunizierende Zustandsmaschinen**

Hydraulischer Aktuator

3 Betriebsmodi:

- Standby
  - Schaltventil wirkt als Last
  - Regelventil läuft mit
- Aktiv
  - Schaltventil offen
  - Regelventil läuft
- Abgeschaltet
  - Schaltventil wirkt als Last
  - Regelventil steht still

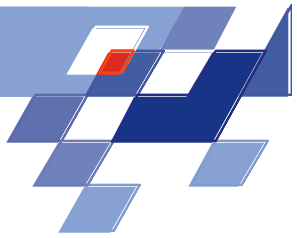




Modellkomponenten

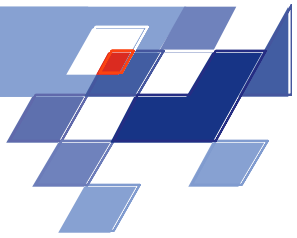
- Flugzeugmodell
  - großes DAE-System
- Hydraulische Komponenten
  - geschaltete Dynamik (diskrete Steuersignale)
  - steife Dynamik (Kompressibilität)
- Ereignis-diskrete Steuerung
  - komplexe diskrete Dynamik
  - sehr viele diskrete Ereignisse (Brute-Force-Ansatz)

➔ **Zielsetzung:**
- **leistungsfähige Modellierung**
- **effiziente hybride Simulation**



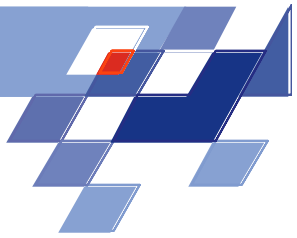
Übersicht

- Beispiel: Höhenrudersystem
- **Anforderungen & Stand der Technik**
- Problemlösung Teil 1:
  - objektorientierte Modellierung, Modelica
- Problemlösung Teil 2:
  - Integration ereignis-diskreter Modelle
- Zusammenfassung und Ausblick



Anforderungen (1)

- Komfortable Beschreibung physikalischer Systeme
  - Standard-Fälle:
    - Verknüpfen vordefinierter Bausteine analog zu Schaltplänen
    - i.d.R. keine festgelegten Ein- und Ausgänge → *akausal*
  - spezielle Komponenten:
    - Gleichungen einfach „abtippen“ → *gleichungsbasiert*
    - ohne Sortieren und Auflösen → *deklarativ*
  - Übersicht: Hierarchische Modellstruktur
    - Aggregation von Komponenten zu größeren Bausteinen
  - Unterstützung benutzerdefinierter Baustein-Bibliotheken
    - Wiederverwendbarkeit



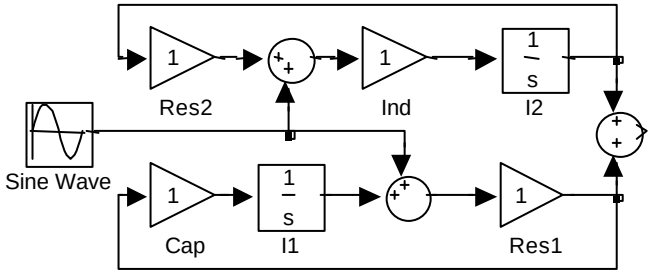
Anforderungen (2)

- Integration diskreter Komponenten (Steuerungen)
  - Verwendung problemspezifischer Formalismen
    - Automaten, Statecharts, Petri-Netze, ...
    - Datenflußdiagramme, synchrone Sprachen, ...
    - SPS-Programmiersprachen (Sequential Function Charts, FBS)
  - Interoperation verschiedener Formalismen
  - Nahtlose Integration in physikalische Modelle
    - intuitive Verknüpfung mit physikalischen Komponenten
    - Synchronisation mit Integrator über Zustandsereignisse
    - korrekte Ausführung des diskreten Verhaltens
  - Anschauliche Visualisierung der diskreten Abläufe

Stand der Technik

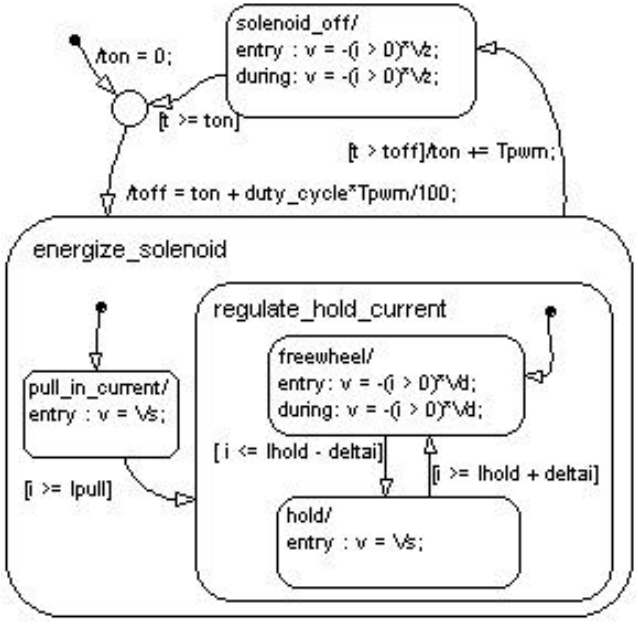
Blockdiagramme

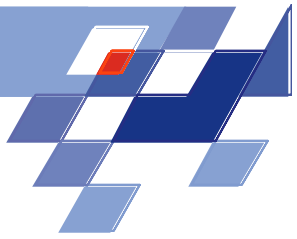
- abstrakte mathematische Sicht
- kausale Schnittstellen
- keine DAE-Solver



Stateflow

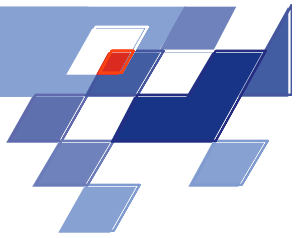
- Statechart-Variante
- nicht für alle Anwendungen geeignet
- unsaubere Synchronisation diskreter und kontinuierlicher Simulation





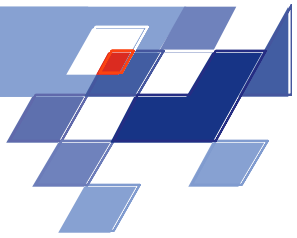
Übersicht

- Beispiel: Höhenrudersystem
- Anforderungen & Stand der Technik
- **Problemlösung Teil 1:**
  - **objektorientierte Modellierung, Modelica**
- Problemlösung Teil 2:
  - Integration ereignis-diskreter Modelle
- Zusammenfassung und Ausblick



Objektorientierte Modellierung

- Vorteile
  - deklarative, gleichungsbasierte Modellierung von Bausteinen
  - akausale Schnittstellen
  - physikalische Struktur des Systems wird abgebildet
  - symbolische Transformation des Gesamtgleichungsystems
    - ➔ effiziente Simulation
  - DAE-Löser sind Standard
- Nachteil
  - ungeeignet für Modellierung komplexer diskreter Systeme
    - Bausteinstruktur oft untauglich
    - Gleichungskonzept erlaubt keine lokalen Iterationen



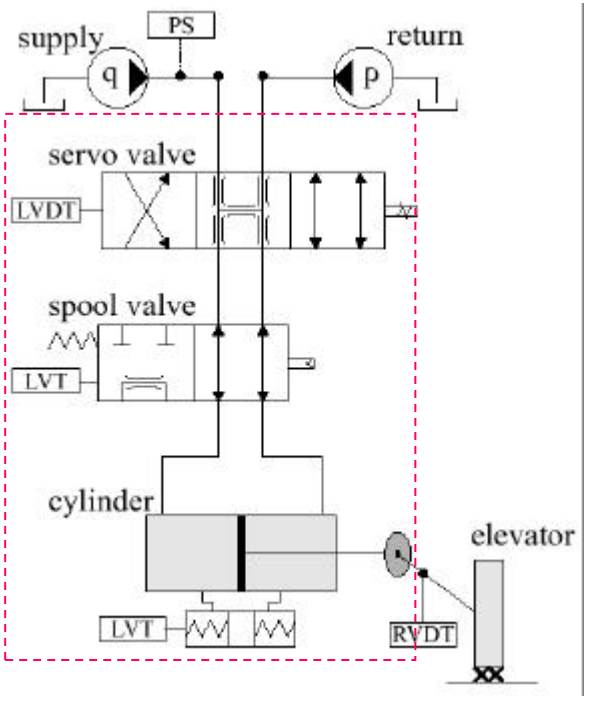
Spezifische Modelica-Vorteile

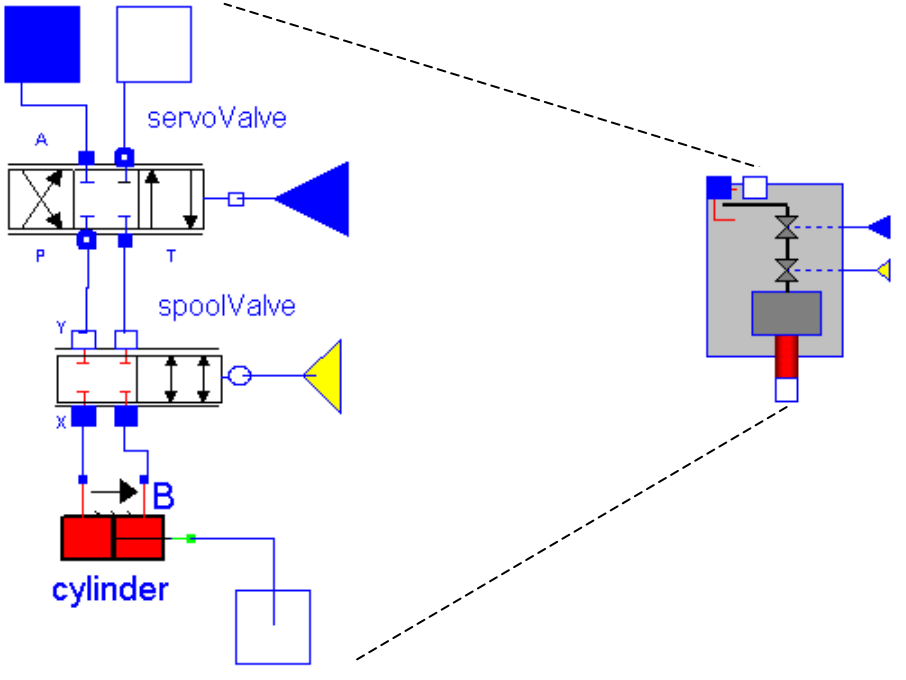
- standardisiert und frei verwendbar
- nicht auf bestimmte Anwendungen zugeschnitten
- verschiedene Bibliotheken kombinierbar
- Klassenkonzept erleichtert Bibliotheksaufbau

Software-Werkzeug Dymola:

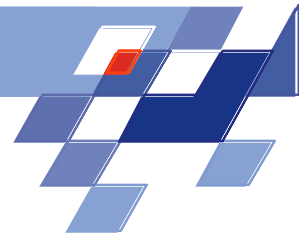
- grafische Modellierung physikalischer Systeme
- leistungsfähige symbolische Transformation
- geschaltete (implizite) DAEs können simuliert werden

Modellieren mit Modelica



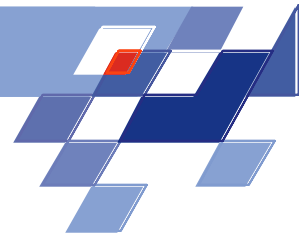


Technische Beschreibung → Modelica-Modell in Dymola → Aggregierter Baustein



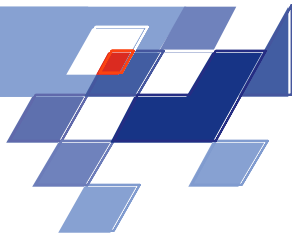
Übersicht

- Beispiel: Höhenrudersystem
- Anforderungen & Stand der Technik
- Problemlösung Teil 1:
  - objektorientierte Modellierung, Modelica
- **Problemlösung Teil 2:**
  - **Integration ereignis-diskreter Modelle**
- Zusammenfassung und Ausblick



Grundkonzept

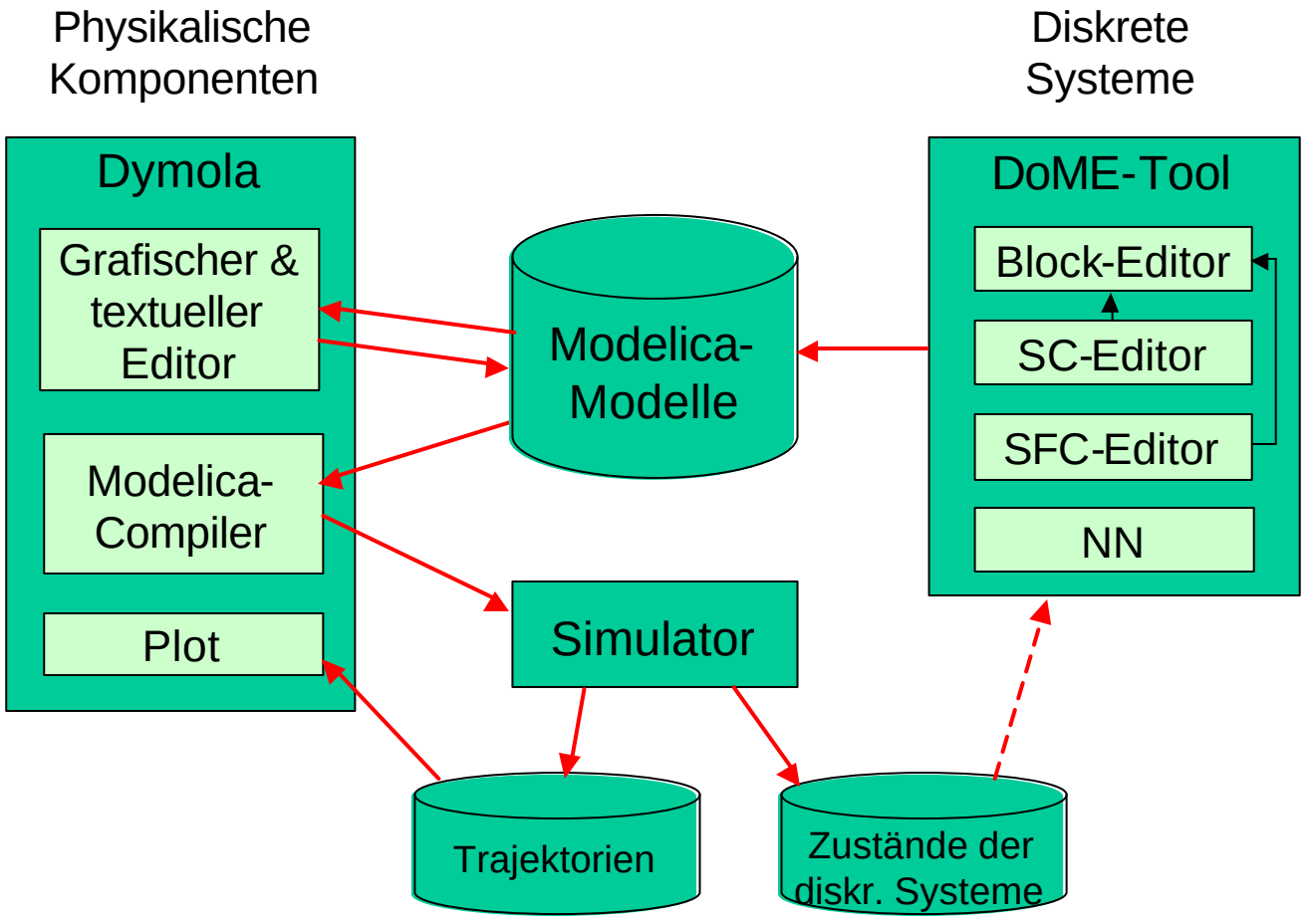
- Modellierungsumgebung für ereignis-diskrete Systeme
  - Keine syntaktischen und grafischen Einschränkungen
  - Heterogenität und Hierarchie (spez. Blockgraph-Formalismus)
  - Archetyp-Konzept für Wiederverwendung und Konfiguration
  - Meta-Modell-Ansatz gestattet einfache Einbettung weiterer Formalismen (bereits realisiert: Statecharts, SFCs)
- Aus diskretem Modell wird Modelica-Baustein generiert
  - intuitive Verknüpfung mit physikalischen Komponenten
- Verhalten durch Modelica-Algorithmus definiert
  - Keine semantischen Einschränkungen

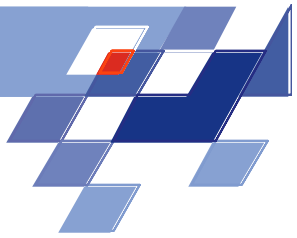


Meta-Modell-Ansatz

- Modellierung
  - Werkzeug: DoME (Honeywell), frei verfügbar
  - Generieren von grafischen Editoren aus Meta-Modell:
    - Formale Spezifikation der Syntax + Grafikparameter
- Übersetzung
  - Teilmodelle werden in Standardform übersetzt:
    - Datenstruktur + Algorithmus als abstrakter Syntaxbaum
  - Automatisch verfügbare Funktionalität:
    - Interoperation mit bestehenden Formalismen
    - Übersetzung nach Modelica
    - Protokollierung & Einlesen der Simulationsdaten und Navigation
    - Animation der grafischen Beschreibung

Modellierungs- & Simulationsumgebung

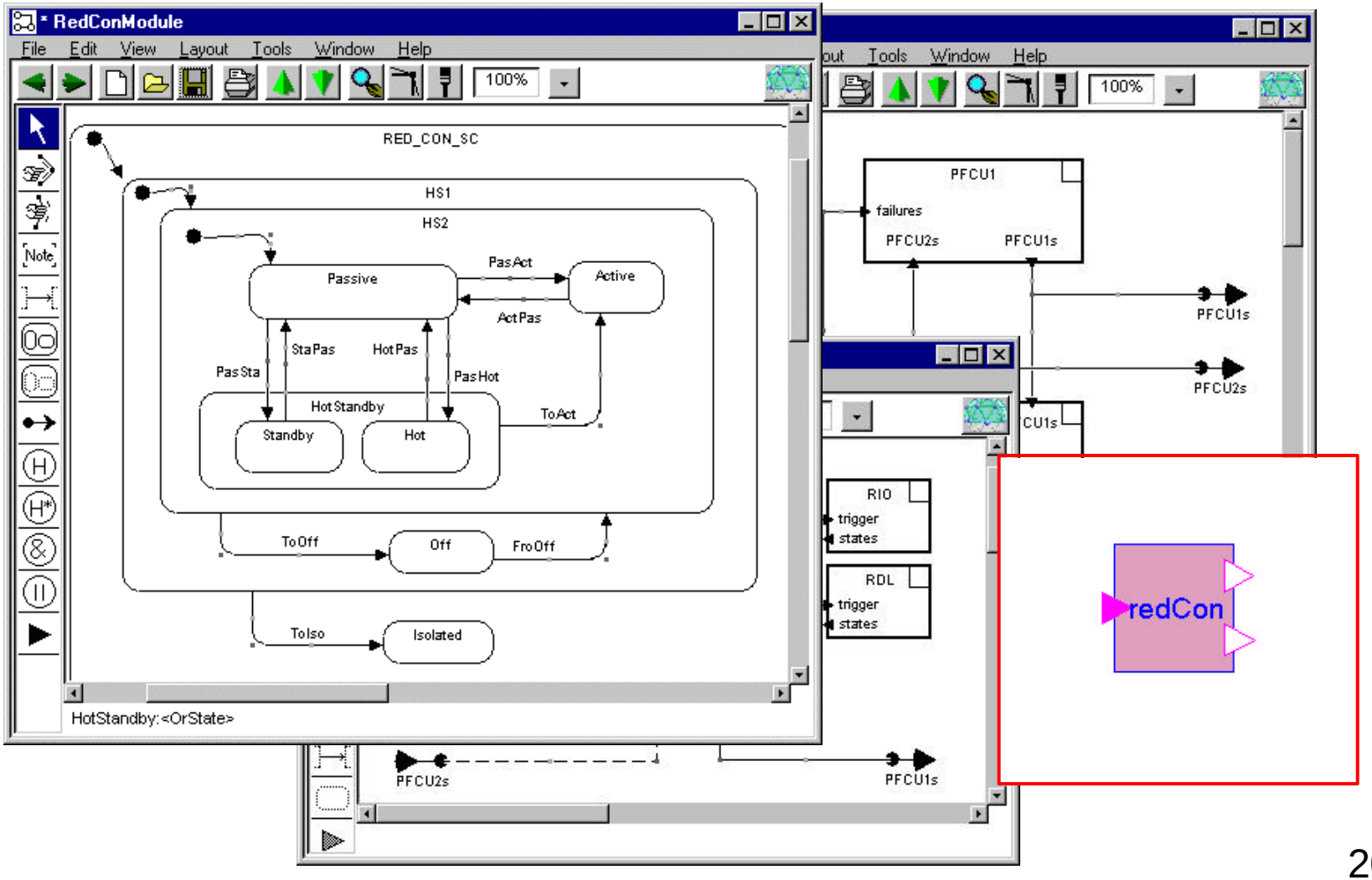




Synchronisation

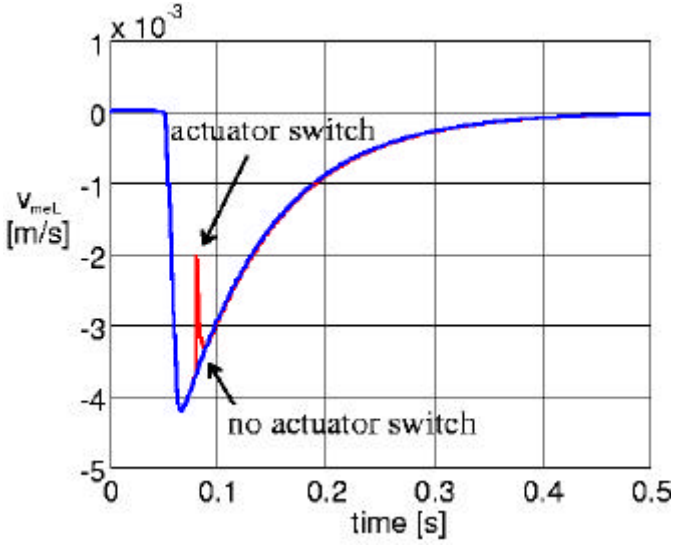
- Schaltbedingungen in Formalismen treten im generierten Algorithmus als logische Ausdrücke auf
- Ungleichungsausdrücke definieren Schwellwerte im physikalischen System.
- Modelica-Compiler nutzt Ungleichungsausdrücke für die Erkennung von Zustandsereignissen
  - Zustandsereignisse werden detektiert und lokalisiert
  - Integration wird unterbrochen
  - Diskrete Aktionen werden zum richtigen Zeitpunkt ausgeführt!

Modellierung der diskreten Steuerung

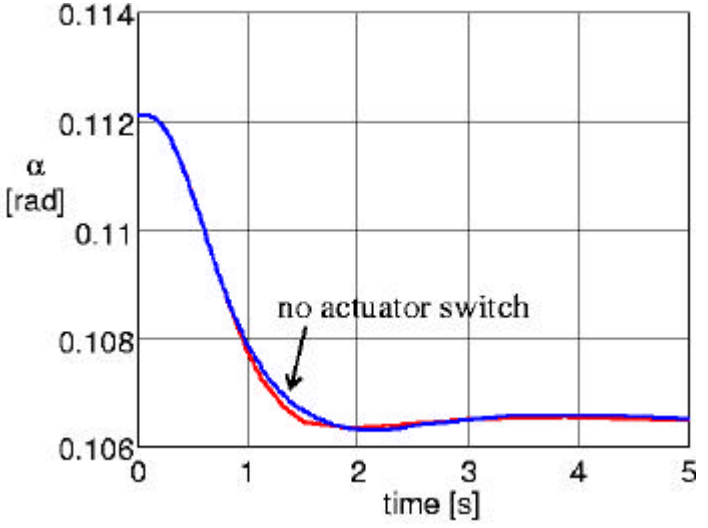


Simulationsergebnisse

Mit und ohne Ausfall eines Aktuators

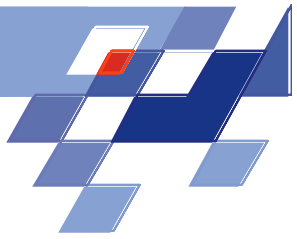


Höhenruder-Geschwindigkeit



Flugzeug-Neigung

Simulationsdauer im Sekundenbereich, über 2300 Variablen



Zusammenfassung und Ausblick

- Physikalische und ereignis-diskrete Modelle stellen sehr unterschiedliche Anforderungen an Modellierung und Simulation
  - objekt-orientiert, deklarativ, gleichungsbasierung
  - Spezielle Editoren, imperativ, algorithmisch
- Effiziente Simulation
- Modelica-kompatibel ➔ Modellaustausch möglich

Ausstehende Arbeiten

- Umsetzung der Visualisierung von Simulationsläufen
- Implementierung weiterer Formalismen
- engere Verzahnung von SFCs und SCs