

# Einführung in Open Modelica



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

TU Darmstadt, Fachgebiet Fluidsystemtechnik

## Allgemeines zu OpenModelica (OM)

### Bedienung von OpenModelica

- Benutzeroberfläche OM-Notebook
- FreeModelicaEditor (FME)
- Rechnen von einfachen Beispielen

### Programmierung von Beispielsystemen

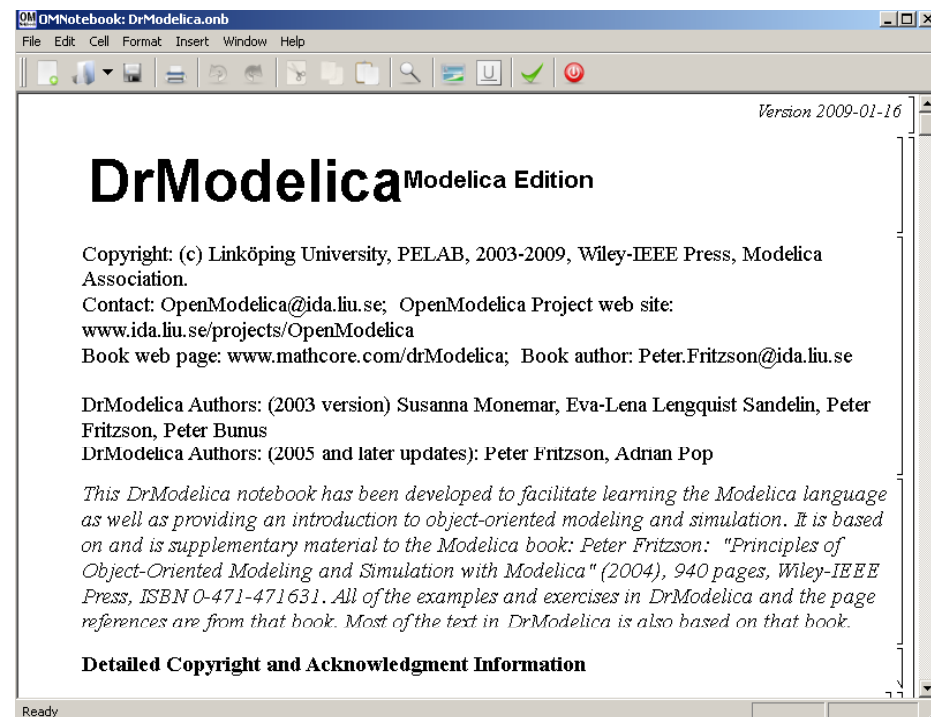
- Komponentenprüfstand (servohydraulische Prüfmaschine)
- Tanksystem

# Was ist Modelica

- Modelica := Sprachstandard zur Objektorientierten Modellierung physikalischer Systeme
- Distributionen:
  - Kommerzielle Anbieter: Dymola, SimulationX, AMESim, MathModelica
  - Open Source: **OpenModelica**, SimForge
- Vorteile gegenüber Matlab-Simulink:
  - Leichte Physikalische Modellbildung durch objektorientierten Modellaufbau
  - Zahlreiche existierende Bibliotheken: Mechanik, Pneumatik, Hydraulik, E-Technik, ...
  - Erweiterbarkeit der Modelle: White-Box-Modellcharakteristik
  - Schnelle Modellbildung durch direkte Eingabe von Differentialgleichungen und algebraischen Gleichungen

# OpenModelica OM

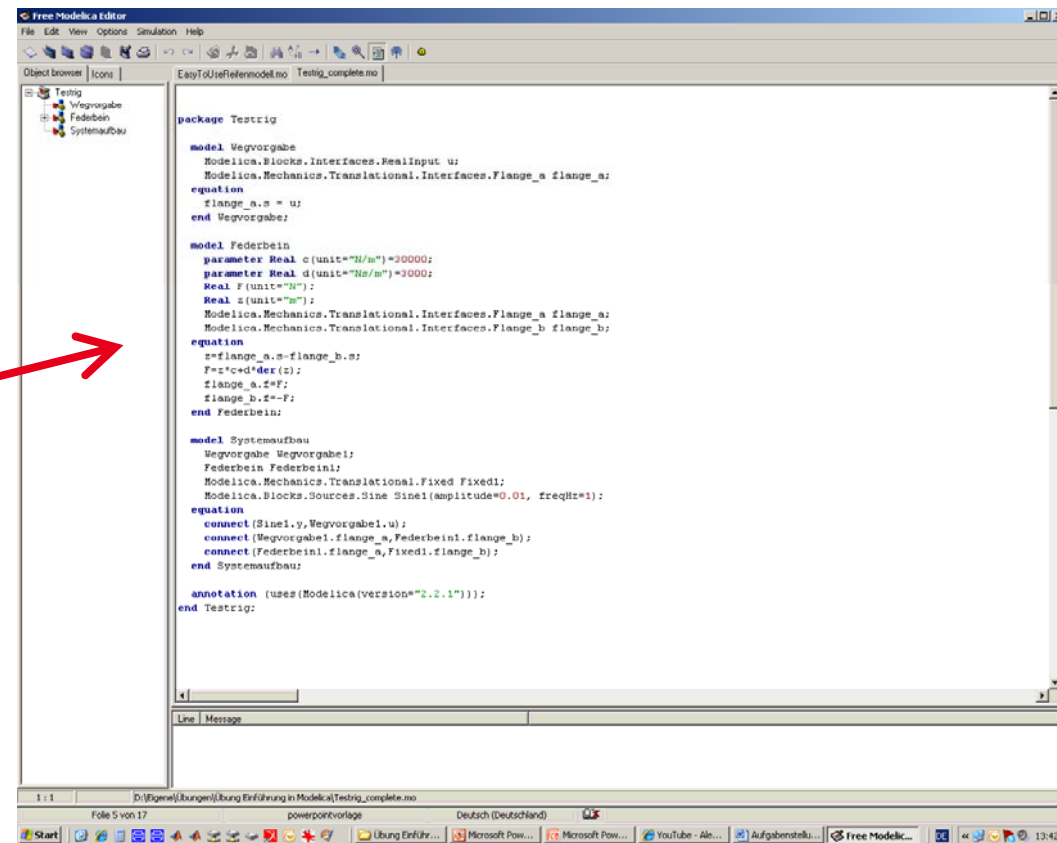
- OpenModelica ist eine freie Distribution von Modelica
- OMNotebook (Dr.Modelica) ist die zugehörige textbasierte Oberfläche
- Features von OpenModelica:
  - Simulation
  - Plotten der Ergebnisse
- Free Modelica Editor (FME) zur Erstellung von Modelica Files



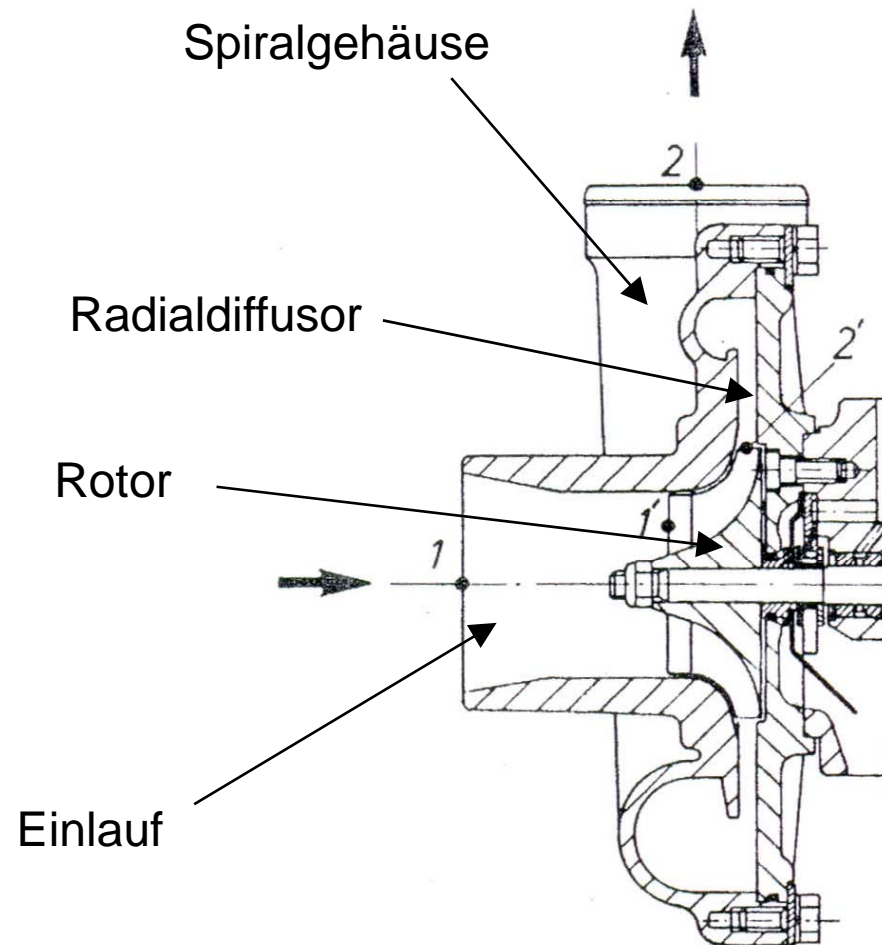
# Free Modelica Editor (FME)

Modelica Code  
Erstellung prinzipiell in  
jedem Editor möglich.

Vorteil des FME: Syntax  
Highlighting



# 1. Schritt: Zerlegung des Systems in einzelne Komponenten



Durch die Hintereinanderschaltung der Einzelkomponenten entsteht das Gesamtmodell.

## 2. Schritt: Annahmen treffen

- Stationäre Strömung
- Adiabate Strömung
- Keine Abhängigkeit der spezifischen Wärmekapazitäten  $c_p$  und  $c_v$  von der Temperatur
- Ideales Gas
- Isentrope Zustandsänderung
- 1-dimensionale Turbomaschinentheorie (Stromfadentheorie)

### 3. Schritt: Gleichungen aufstellen

- Energiebilanz

$$h_{t,a} = h_{t,e} + w_{ea} + q_{ea}$$

- Kontinuitätsgleichung

$$\dot{m}_a = \dot{m}_e$$

- Drallsatz (Ringdiffusor und Rotor)

$$M = \dot{m}(r_a c_{ua} - r_e c_{ue})$$

- Ideale Gasgleichung

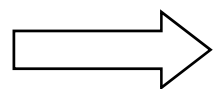
$$\frac{p}{\rho} = RT$$

- Zustandsgleichungen, z. B.

$$h = c_p(T - T_0) + h_0$$

- Isentropenbeziehung

$$\frac{p}{\rho^\gamma} = \text{const}$$

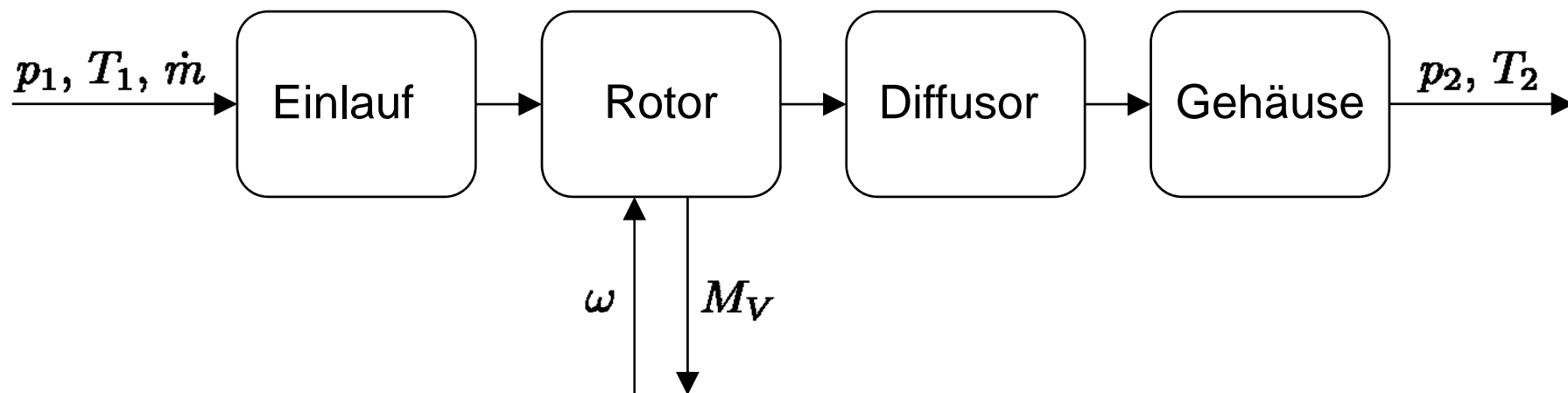


Anwendung der Gleichungen auf die Teilmodelle



## 4. Schritt: Umsetzung in Modelica-Code

- N Gleichungen und N Unbekannte



# Allgemeine Programmstruktur

```
model Testprogramm
```

```
parameter Real laenge=5.3;  
parameter Real breite=1.2;  
parameter Real hoehe=1.2;
```

```
Real A;  
Real V;
```

```
equation
```

```
A=laenge*breite;  
V=A*hoehe;
```

```
end Testprogramm;
```

- Allgemeiner Aufbau eines Modells: **model**, **equation** und **end**
- Parameter und Variablen definieren
- Semikolon nach jeder Zeile!
- Nach **equation** erfolgt Eingabe der Gleichungen. Reihenfolge der Eingabe spielt keine Rolle!
- **N Gleichungen und N Variablen. Wichtig!!!**

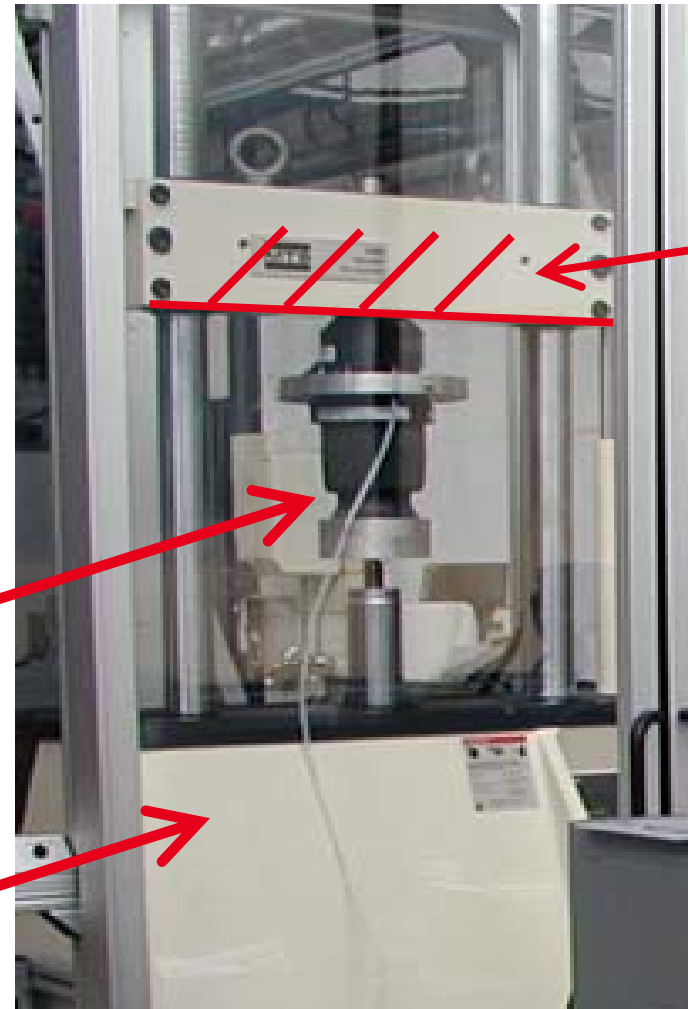
# Beispiel Komponentenprüfstand



Regler zur  
Sollwertvorgabe

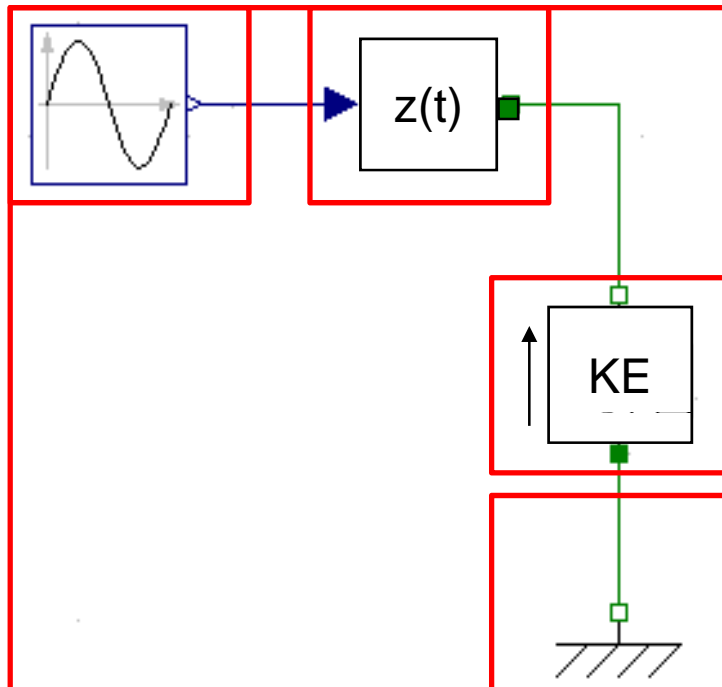
Kraftelement  
z.B. Luftfeder

Servohydraulische  
Prüfmaschine



Festlager

# Umsetzung in OpenModelica



```
package Testrig
```

```
model Wegvorgabe
  Modelica.Blocks.Interfaces.RealInput u;
  Modelica.Mechanics.Translational.Interfaces.Flange_a flange_a;
equation
  flange_a.s = u;
end Wegvorgabe;
```

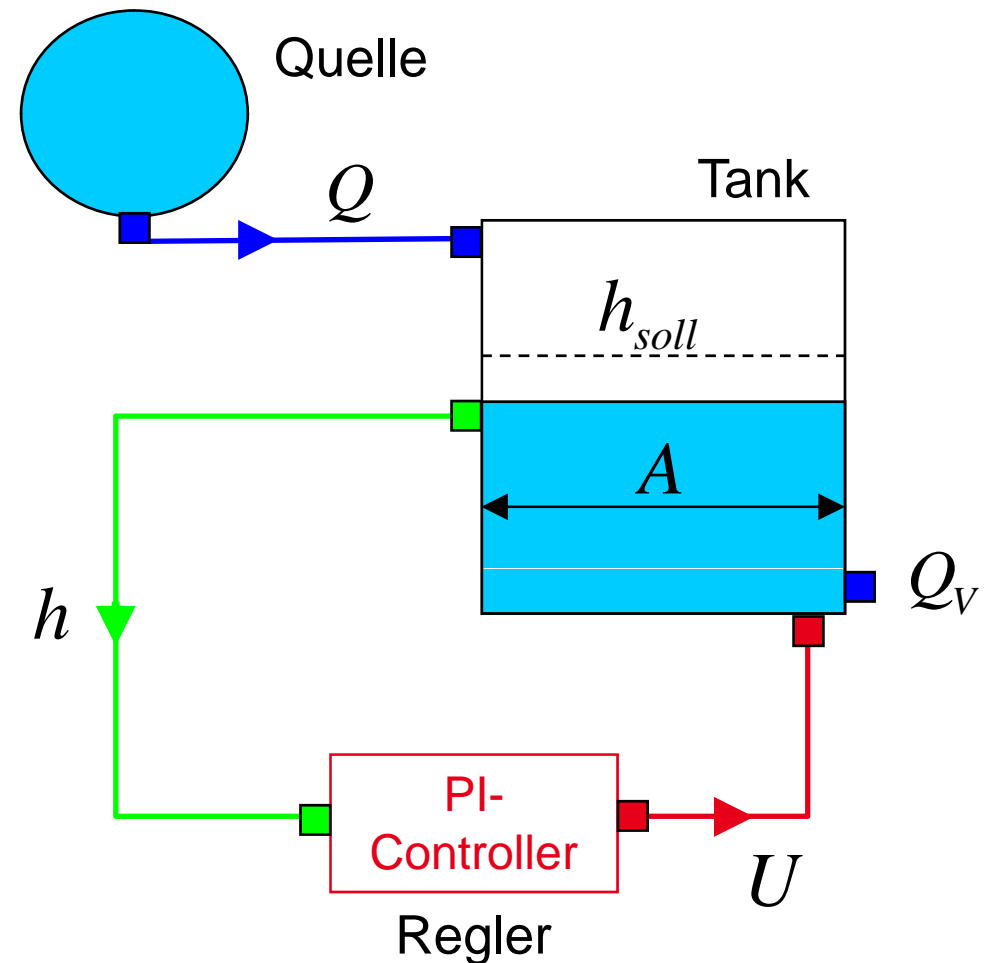
```
model Federbein
  parameter Real c(unit="N/m")=30000;
  parameter Real d(unit="Ns/m")=3000;
  Real F(unit="N");
  Real z(unit="m");
  Modelica.Mechanics.Translational.Interfaces.Flange_a flange_a;
  Modelica.Mechanics.Translational.Interfaces.Flange_b flange_b;
equation
  z=flange_a.s-flange_b.s;
  F=z*c+d*der(z);
  flange_a.f=F;
  flange_b.f=-F;
end Federbein;
```

```
model Systemaufbau
  Wegvorgabe Wegvorgabe1;
  ...
  Modelica.Mechanics.Translational.Fixed Fixed1;
  Modelica.Blocks.Sources.Sine Sine1(amplitude=0.01, freqHz=1);
equation
  connect(Sine1.y,Wegvorgabe1.u);
  connect(Wegvorgabe1.flange_a, ... );
  connect( ... ,Fixed1.flange_b);
end Systemaufbau;
```

```
end Testrig;
```

# Beispiel Tanksystem

- Eine Quelle erzeugt einen Volumenstrom  $Q$  und leitet diesen in einen Tank.
- Erfassung des Höhenstands  $h$  im Tank und Weiterleitung an einen PI-Regler.
- Regler vergleicht Isthöhe mit Sollhöhe und regelt das Auslassventil derart, dass Regelfehler null wird.



# Gleichungen Tanksystem, alle Werte in SI-Einheiten

- Quelle:

$$\text{flowrate} = 0.02$$

$$Q = \text{flowrate} \text{ für } t < 150s$$

$$Q = 3 \text{flowrate} \text{ für } t \geq 150s$$

- Flüssigkeitshöhe im Tank:

$$\dot{h} = \frac{Q - Q_v}{A}$$

$$Q_v = 0.05 U$$

$$A = 1$$

- PI-Regler:

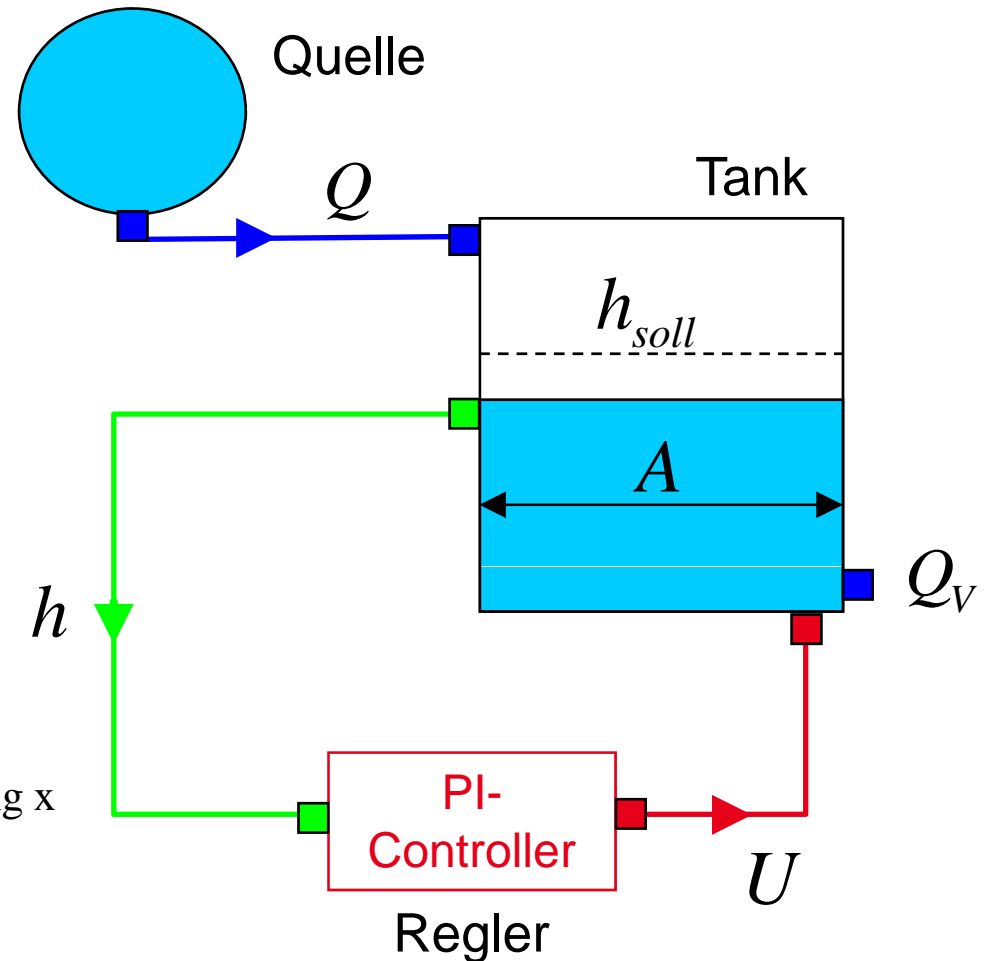
$$\text{error} = h - h_{soll} \quad \text{Regelabweichung}$$

$$\dot{x} = \text{error} / T \quad \text{Integrale Regelabweichung } x$$

$$U = K(\text{error} + x) \quad \text{Reglerausgangssignal}$$

$$K = 2, \quad T = 10 \quad \text{Reglerparameter}$$

$$h_{soll} = 0.25$$



# Vollständiges Tanksystem in einem model



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

```
model Tanksystem_EinModell

parameter Real flowrate=0.02; // Volumenstrom aus der Flüssigkeitsquelle
parameter Real area=1; // Oberfläche des Flüssigkeitsspiegels im Tank
parameter Real flowGain=0.05; // Skalierungsparameter für den aus dem Tank austretenden Volumenstrom
parameter Real K=2; // Verstärkungsfaktor des Reglers
parameter Real T=10; // Zeitkonstante des Reglers (Integraler Anteil)
parameter Real ref=0.25; // Sollwert des Höhenstands der Flüssigkeit im Tank

Real h; // Höhenstand der Flüssigkeit im Tank (Istwert)
Real sensorsignal; // Sensorsignal zur Erfassung des Höhenstands im Tank (Istwert)
Real q_Source; // Aus der Flüssigkeitsquelle austretender Volumenstrom
Real q_Tank_in; // In den Tank einfließender Volumenstrom
Real q_Tank_out; // Aus dem Tank ausfließender Volumenstrom

Real error; // Regelabweichung zwischen Soll- und Istwert des Höhenstands im Tank
Real controller_out; // Ausgangssignal des Reglers
Real x; // Integrale Regelabweichung

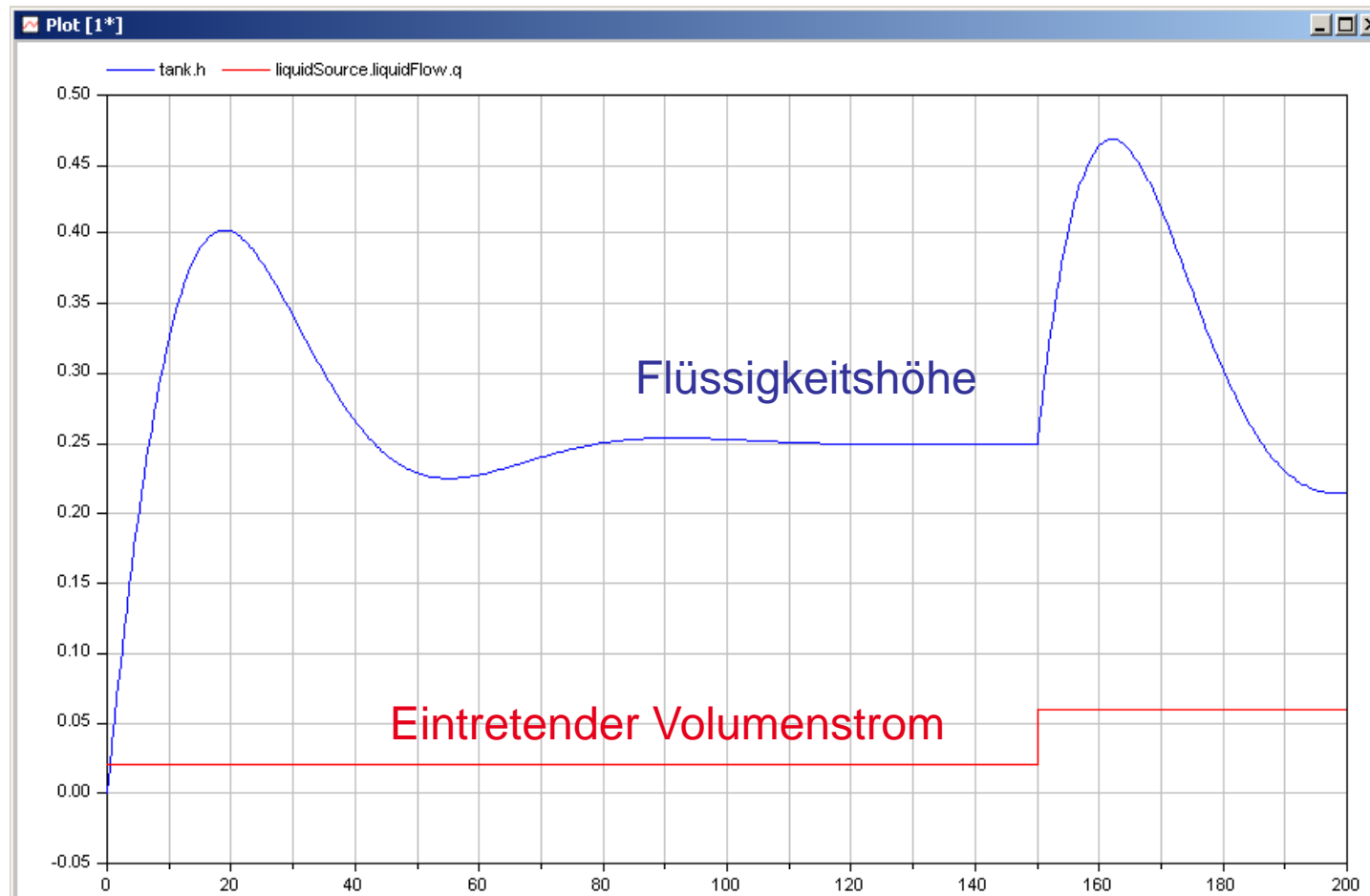
equation
  q_Source = if time < 150 then flowrate else 3*flowrate; // Definition des Volumenstroms aus der Quelle
  q_Tank_in=q_Source; // Volumenstrom in den Tank ist gleich Volumenstrom aus der Quelle
  der(h)=(q_Tank_in-q_Tank_out)/area; // Berechnung des Flüssigkeitsstands im Tank

  sensorsignal=h; // Zuweisung Sensor erfasst Höhenstand im Tank
  error=ref-sensorsignal; // Berechnung der Regelabweichung Sollwert - Istwert
  der(x)=error/T; // Berechnung der integralen Regelabweichung x
  controller_out=K*(error+x); // Ausgangsspannung des Reglers

  q_Tank_out=-flowGain*controller_out; // Ausgangsspannung des Reglers wird umgesetzt in ausfließenden Volumen

end Tanksystem_EinModell;
```

# Simulationsergebnisse





# Häufig verwendete Befehle und Syntax

- Simulationszeit **time**:  $F = F_{\max} * \sin(\omega * \text{time});$
- **If-Then** Abfrage:  $x = \text{if } F < 10 \text{ then Wert1 else Wert2};$
- Definition von **Ableitungen** nach der Zeit:  $x_{\dot{}} = \text{der}(x);$   
 $x_{\ddot{}} = \text{der}(x_{\dot{}});$
- Beispiel Impulssatz:  $m * x_{\ddot{}} + d * x_{\dot{}} + c * x = F;$
- Auskommentieren oder Kommentar: `// ...`
- Weitere Keywords: <http://www.modelica.org/documents/ModelicaSpec30.pdf>

algorithm	and	annotation	assert	block
break	class	connect	connector	constant
constrainedby	der	discrete	else	elseif
elsewhen	encapsulated	end	enumeration	equation
expandable	extends	external	false	final
flow	for	function	if	in
initial	inner	input	loop	model
not	or	outer	output	package
parameter	partial	protected	public	record
redeclare	replaceable	return	then	true
type	when	while	within	

# Literatur und Links zu Modelica

- Tutorial und Buch:  
<http://www.ida.liu.se/labs/pelab/modelica/OpenModelica.html>
- Download von Modelica Distributionen:
  - Dymola (demo version) <http://www.dynasim.se/dymola.htm>
  - OpenModelica (full) <http://www.ida.liu.se/labs/pelab/modelica/OpenModelica.html>
  - MathModelica (demo version) <http://www.mathcore.com/products/mathmodelica/>
  - SimForge (full, requires OM) <http://trac.elet.polimi.it/simforge/trac.cgi/downloader>