Dep. of Automatic Control
Lund Institute of Technology

Modeling of Thermo-Hydraulic Systems with Modelica

Hubertus Tummescheit
Department of Automatic Control
Lund Institute of Technology
Box 118, SE-22100 Lund, Sweden
email: hubertus@control.lth.se

Contents

The Modelica Library Effort

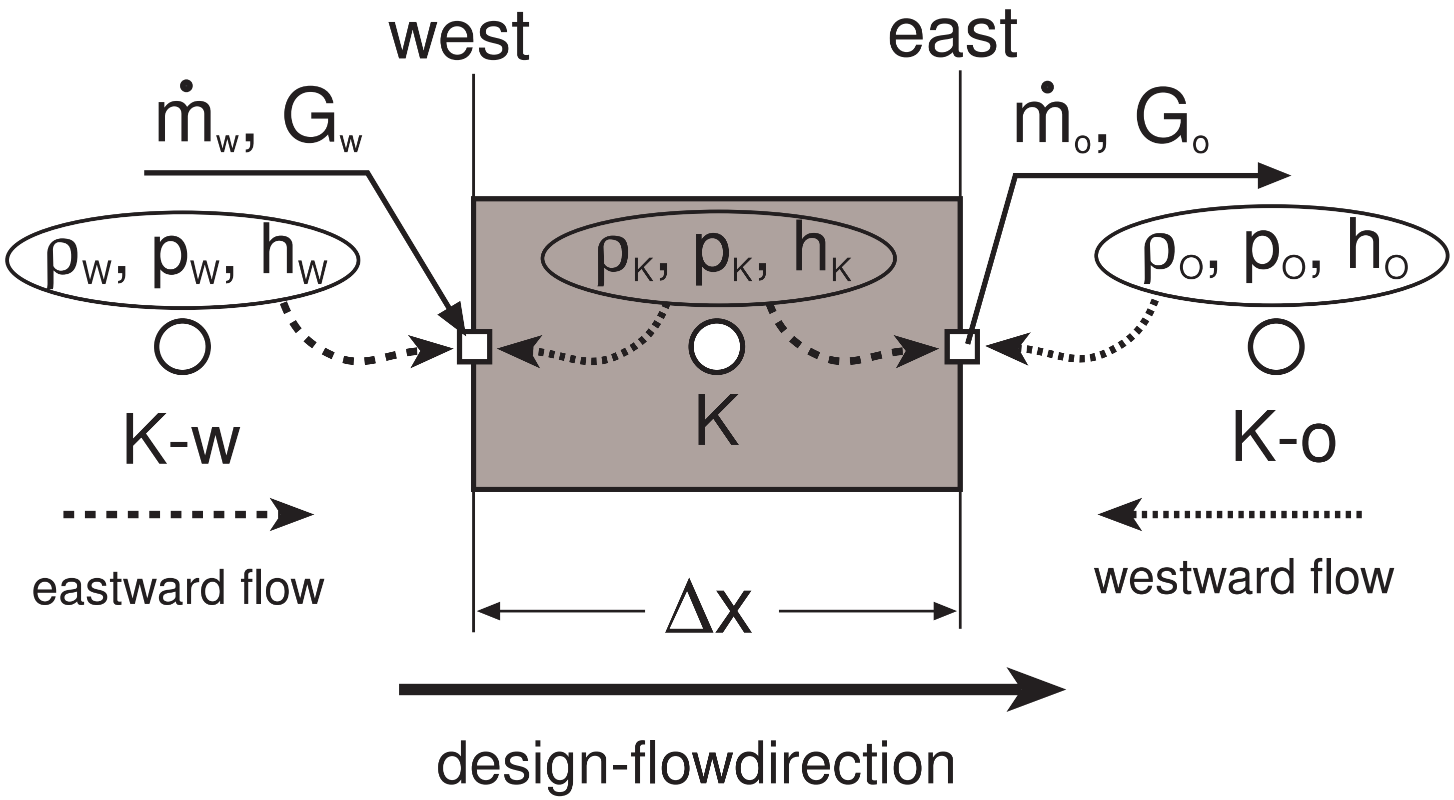
- free basic libraries from many domains available
- Lund University works on library for thermo-hydraulic systems
- other domain libraries:
  - mechanics
  - hydraulics
  - circuits
  - control

Analysis of user requirements

- ‘Complete’ set of application specific component models
- Parameterization easy to handle (geometry, materials)
  - heat exchangers often need a lot of data input.
  - often difficult to map complex geometries onto simple parameter sets.
- Efficient simulation!
- Fast learning: simple library, few models with powerfull parameterization.
  - Analysis of physics and
  - symbolic preprocessing

  can help to reduce number of classes.

Discretization Scheme for Control Volumes



Interface east: $\{\dot{m}_o, p_{K\text{-}o}, \{h_K, G_K, \rho_K\}, \{h_{K\text{-}o}, G_{K\text{-}o}, \rho_{K\text{-}o}\}\}$

Interface west: $\{\dot{m}_w, p_K, \{h_{K\text{-}w}, G_{K\text{-}w}, \rho_{K\text{-}w}\}, \{h_K, G_K, \rho_K\}\}$

Basic Balance Equations

Choose variables for the state equations, e. g. pressure and enthalpy. Formulate as general as possible.

$$p = p(m, e, V) \qquad h = h(m, e, V)$$

$$\frac{dp}{dt} = \left(\frac{\partial p}{\partial m}\right)_{e,V} \frac{dm}{dt} + \left(\frac{\partial p}{\partial e}\right)_{m,V} \frac{de}{dt} + \left(\frac{\partial p}{\partial V}\right)_{m,e} \frac{dV}{dt}$$

$$\frac{dh}{dt} = \left(\frac{\partial h}{\partial m}\right)_{e,V} \frac{dm}{dt} + \left(\frac{\partial h}{\partial e}\right)_{m,V} \frac{de}{dt} + \left(\frac{\partial h}{\partial V}\right)_{m,e} \frac{dV}{dt}$$

$$\left(\frac{\partial p}{\partial m}\right)_{e,V} = \frac{\rho + \left(\frac{\partial \rho}{\partial h}\right)_p h}{\left(\frac{\partial \rho}{\partial p}\right)_h \rho + \left(\frac{\partial \rho}{\partial h}\right)_p}$$

etc.

The Momentum Balance

$$\mathbf{dynamic}\left[\Delta z\frac{d\dot{m}}{dt} - \Delta\left(\dot{m}w\right)\right] = \Delta p A + \mathbf{gravity}\left[g_z\rho V\right] - F_R$$

Boolean variables as switches for model assumptions:
dynamic chooses a stationary or dynamic momentum balance.
gravity sets the assumption, if a gravity head is used.

Symbolic preprocessing removes terms like

$$0\left[g_z\rho V\right]$$

from the equations for improved efficiency

Medium Properties

different ways supported by Modelica:

- Modelica is well suited to write medium property models: a highly optimized version of the newest standard for steamtables, IAPWS/IF97, has been implemented in Modelica.
- External function interface allows to reuse any existing property routines, preferably those that are designed for dynamic simulation. Fortran and C are supported languages for calling external code. Used for refrigeration properties, NIST REFPROP.

Medium Properties

Efficient dynamic simulation requires:

- derivatives with respect to the chosen state variables
- 1- and 2-phase regions implemented as distinct regions
- ancillary equations of the phase change border as a function of the state variables.
- symbolic calculation of Jacobian entries.

Library Design

Issues in model reuse

- language issues
- tool issues

Modelica: 3 different means of code reuse

- inheritance, even multiple inheritance
- composition
- generic models via class parameters

Library Design

- Model user view
  - Wants It-Just-Works™ Models
  - Fast & simple in routine use
  - Does not care about how its implemented
- Model developer/maintainer view
  - Cares about structure and reuse
  - Local changes
- Company view
  - What does it cost?
  - Long term risks or advantages.

Library Design

How to map domain knowledge to library design?

- get a good overview over the physical models.
- Try to identify *Design Patterns* in the models: main concepts or components for reuse.

Library Design

What are the main units of reuse in thermo-hydraulics?

- control volume
- separation of medium and machine
- machine characteristic functions
- parameterization: mapping from complex geometries to simple model parameters.

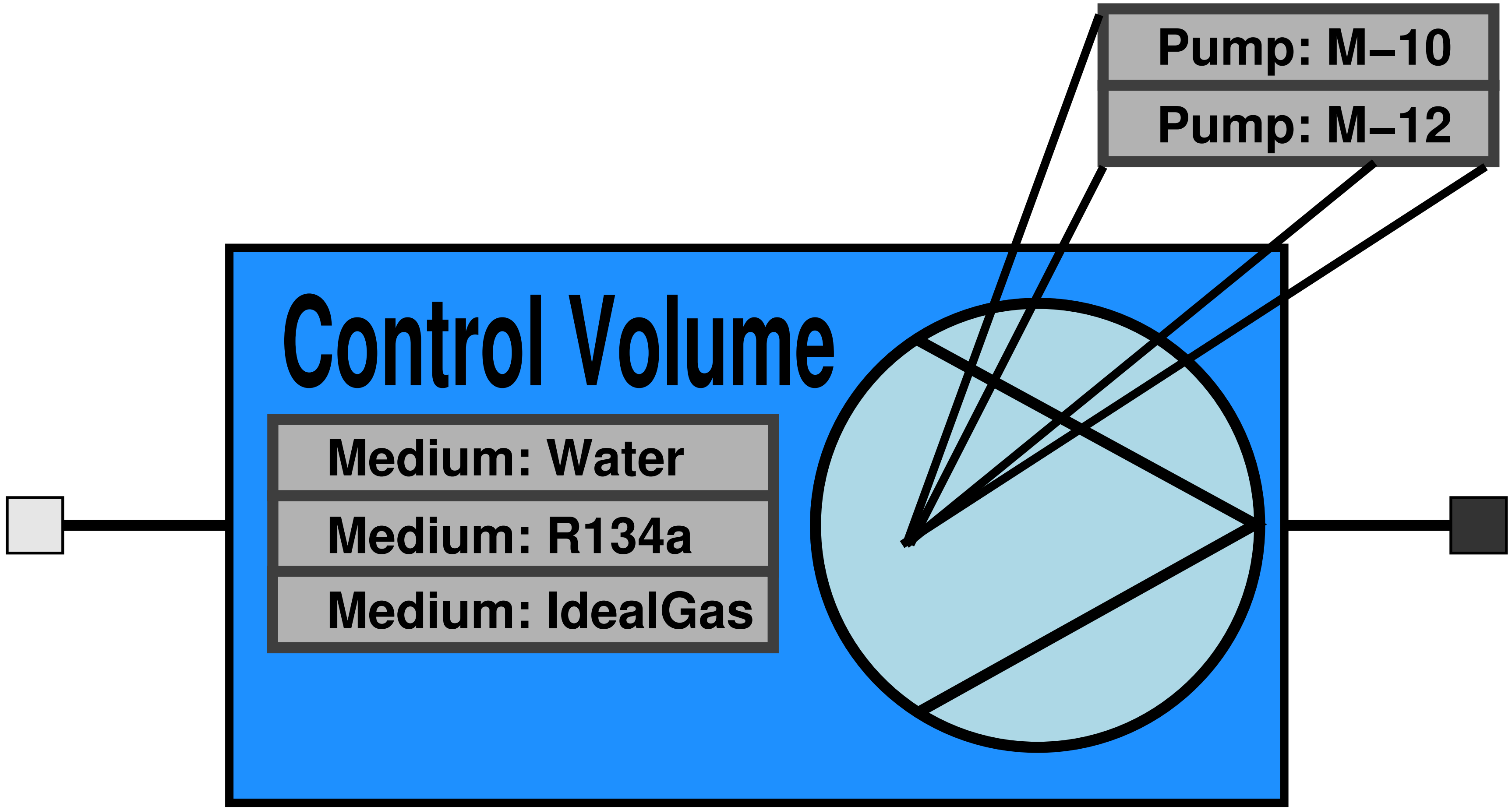
Example: a pump

How can you model a pump in Modelica that works equally well for water as for multi-component mixtures?

- different interfaces
- characteristic functions for pressure vs flow
- Simple and complete or with flange for motor

The complete pump model consists of a control volume that also contains the appropriate interfaces and a pump

Example: a pump



```
model GenericPump
  replaceable model CV=BaseCV_ph;
  extends CV;
  replaceable model PumpEq=SimplePump;
  PumpEq pump;
  equation
  connect(upstream.p,pump.p_upstream);
  connect(downstream.p,pump.p_downstream);
%  connect(upstream.d,pump.d_upstream);
%  connect(downstream.d,pump.d_downstream);
end GenericPump;

model MixPump
  extends GenericPump(;
  redeclare CV=NFlowCV_ph,
  redeclare PumpEq=PumpEqMix);
end MixPump;
```

Simplicity vs Flexibility

- Users don’t want flexibilty beyond their application scope
- Specialized libraries need limited flexibility
- Class parameters make validation more difficult

Conclusion

- Model library design
  - requires thorough domain knowledge
  - compromise between simplicity and flexibility
- Class parameters
  - increase flexibility
  - require good planing
  - can make validation