WaterHub Documentation

Release 0

Bruno Hadengue

Dec 19, 2018

Contents

1 Description **1**
1.1 Userguide 1
1.2 Models 2

CHAPTER 1

Description

The WaterHub Package is a Modelica package loosely inspired from the Modelica.Fluid library. It is meant to contain building blocks for the modeling of household-level water systems. The building blocks are clustered into categories which are all inter-compatible. The user can use a connection editor such as OpenModelica OMEdit to build his custom water system.

- End-Uses
- Supply
- Sinks & Reservoirs
- Recovery Systems
- Pipes & Carriers

This package is still under construction, more to come. . .

1.1 Userguide

This section will take you through a step-by-step implementation of your first model using the WaterHub building blocks.

1.1.1 Installation

The WaterHub Modelica package can be loaded as any other Modelica library. The Standard Modelica Library (3.2.2, maybe others) shoud be loaded alongside, as some dependencies are still present. You can clone the library from GitHub using

```
git clone https://github.com/brunohad/WaterHub.git
```

from the terminal (working directory is the location where you want the repository to be cloned to).

1.2 Models

Private tour through all models contained in the WaterHub package.

1.2.1 EndUses

The EndUses package clusters all appliances within an household that allow interaction with a water consumer. They are of special interest because they are associated with a consumer-generated flow of water that acts as a trigger for the rest of the system's building blocks.

BaseEndUses

Model Name	BasicValve	*Partial Model for the definition of end-uses*
Type	Partial Model	
Inlets	`inletCold`	*cold water inlet*
	`inletHot`	*hot water inlet*
	`flowInput`	*data input from e.g. block HydrographFromFile*
Outlets	`outlet`	*water outlet*

Partial model acting as a basic valve: two water inlets mixing into one water outlet. The flow is triggered by the `flowInput` port.

Showers

Base Shower

Model Name	BaseShower	*Partial Model for showers*
Type	Partial Model	
Parameters	`T_wanted`	*Targeted Temperature*

Base model for showers. It simply makes sure $T_{cold} \leq T_{wanted} \leq T_{hot}$ using a trivial algorithm:

```
algorithm
  if T_wanted < inletCold.T then
    T_achieved := inletCold.T;
  elseif T_wanted > inletHot.T then
    T_achieved := inletHot.T;
  else
    T_achieved := T_wanted;
  end if;
```

Classic Shower

Model Name	ClassicShower	*Lossless shower model*
Type	Model	*extends partial model* `BaseShower` *and partial model* `BasicValve`
Inlets	`inletCold`	*cold water inlet*
	`inletHot`	*hot water inlet*
	`flowInput`	*data input from e.g. block HydrographFromFile*
Outlets	`outlet`	*water outlet*
Parameters	`T_wanted`	*Targeted Temperature*

Simple model for showers. The flow is triggered through the `flowInput` port, connected to e.g. an `HydrographFromFile` block. Energy and mass balance equations describe the thermal behavior:

$$T_{out} m_{out} = T_{in}^{cold} m_{in}^{cold} + T_{in}^{hot} m_{in}^{hot}$$

$$m_{in}^{cold} + m_{in}^{hot} = m_{out}$$

Taps

Model Name	ClassicShower	*Lossless tap model*
Type	Model	*extends partial model* `BasicValve`
Inlets	`inletCold`	*cold water inlet*
	`inletHot`	*hot water inlet*
	`flowInput`	*data input from e.g. block HydrographFromFile*
Outlets	`outlet`	*water outlet*
Parameters	`T_wanted`	*Targeted Temperature*

Analogous to classicshowerref.

1.2.2 RecoverySystems

Models specialized in energy recovery. Heat exchanger and heat pump models are the most obvious examples.

Heat Exchangers

Simple Heat Exchanger

Model Name	SimpleHeatExchanger	*Black box heat exchanger that retrieves energy with user defined efficiency*
Type	Model	
Inlets	`inlet`	*water inlet*
Outlets	`outlet`	*water outlet*
	`heat_out`	*heat flow outlet*
Parameters	`efficiency`	*Efficiency factor for the heat recovery*

Not So Simple Heat Exchanger

Model Name	NotSoSimpleHeatExchanger	*Water-water Heat exchanger model*
Type	Model	
Inlets	`inletCold`	*cold water inlet*
	`inletHot`	*hot water inlet*
Outlets	`outletCold`	*cold water outlet*
	`outletHot`	*hot water outlet*
Parameters	`alphaFactor`	*Efficiency factor for the heat recovery. Depends on length of tube, heat exchange coefficient, flows and contact areas*
	`flowHE`	*1 if parallel flow, -1 if counterflow*

The NotSoSimpleHeatExchanger Model has been inspired by a derivation from the Wikipedia page on heat exchangers, in the section “A model of a simple heat exchanger”. This derivation is based on the Book “Fluid Mechanics and Transfer Processes”, Cambridge University Press, Kay J.M. and Nedderman R.M.

The simplest heat exchanger consist of two straight pipes with fluid flows. Let the pipe be of length L, with fluid capacities C_i, flow rates j_i, and temperature profiles along the pipes $T_i(x)$. Assume the heat transfer occurs only transversely between the two fluids and not along the pipe. From Newton’s law of cooling:

$$\frac{\partial u_1}{\partial t} = \gamma(T_2 - T_1)$$
$$\frac{\partial u_2}{\partial t} = \gamma(T_1 - T_2)$$

where *u_i(x)* is the thermal energy profile. It must be noted here that this is for parallel flows. Counterflows heat exchangers require a negative sign in the second equation. γ is the thermal connection constant, a function of the heat exchange coefficient and the contact area. The time change in thermal energy for a fluid unit volume being carried along the pipe can be also written as

$$\frac{\partial u_1}{\partial t} = C_1 j_1 \frac{\partial T_1}{\partial x}$$
$$\frac{\partial u_2}{\partial t} = C_2 j_2 \frac{\partial T_2}{\partial x}$$

Here, $C_i j_i$ are the thermal flow rates. So, equating above equations results in a steady-state, x-only differential equation, that can be solved with

$$T_1(x) = A - \frac{Bk_1}{k} e^{-kx}$$
$$T_2(x) = A + \frac{Bk_2}{k} e^{-kx}$$

where $k_i = \gamma/(C_i j_i)$, $k = k_1 + k_2$ and A, B being integration constants. Knowing the input temperatures at (x = 0) T_{10} and T_{20}, we can derive (for parallel flows)

$$B = (T_{20} - T_{10})$$
$$= \Delta T$$
$$A = T_{10} + \frac{\Delta T}{(1 + j_1/j_2)}$$
$$= T_{20} - \frac{\Delta T}{(1 + j_2/j_1)}$$

$$T_{1L} =$$

$$T_{10} + \frac{\Delta T}{(1 + j_1/j_2)}(1 - e^{\frac{\gamma}{j_1+j_2}L})$$

$$T_{1L} =$$

$$T_{20} - \frac{\Delta T}{(1 + j_2/j_1)}(1 - e^{\frac{\gamma}{j_1+j_2}L})$$

Letting $\alpha = (1-e^{\frac{\gamma}{j_1+j_2}L})$, this term thus describes the efficiency of the heat-exchanger, depending on many parameters such as the heat exchange coefficients, exchange surface area and length of pipes. With $\alpha = 0$, no heat is transferred between the pipes, while all the available heat is transferred when $\alpha = 1$.

1.2.3 Pipes & Carriers

Models of water pipes, electric wires and other carrier systems.

Water Pipes

PipeLossesAtRest

Model Name	PipeLossesAtRest	*Loses energy to environment when water flow is zero*
Type	Model	
Inlets	`inlet`	*water inlet*
Outlets	`outlet`	*water outlet*
	`heatOutlet`	*heat flow outlet*
Parameters	`triggerValue`	*Triggers modeling of heat losses when flow get below*
	`pipeLength`	*Length of water pipe in meters*
	`pipeDiameterInside`	*Inside diameter of pipe in meters*
	`pipeThickness`	*Thickness of pipe walls in meters*
	`material`	*Roll menu to choose material thermal properties*
	`tEnvironment`	*Temperature of external air in Kelvin*

Model of a water pipe that loses energy to its environment when the flow is zero (i.e when water is stagnating in the pipe). The model computes the *UA*-value, i.e. the total thermal conductance of the pipe, using:

$$\frac{1}{UA} = \frac{1}{h_{ci}A_i} + \sum \frac{s_n}{k_n A_n} + \frac{1}{h_{co}A_o}$$

where h_{ci} and h_{co} are the convection heat transfer coefficients of the inside, respectively outside fluid. A_i and A_o are the inside, respectively outside contact areas. s_n is the thickness, k_n the thermal conductivity and A_n the mean area of pipe layer *n*.

UA is then used in the ODE:

$$VC_v\frac{dT}{dt} = -UA(T - T_{env})$$

to compute the time-dependent fluid temperature.