Simulink Report: Valve_

Christian Müller

2008-03-06

Model - Valve_

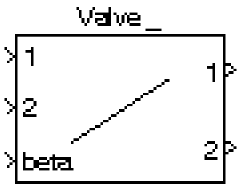


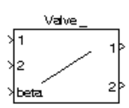
Tabelle 1.1. Valve_ Simulation Parameters

Solver ode14x	*ZeroCross* on	*StartTime* 0.0*StopTime* 10.0
RelTol 1e-3	*AbsTol* auto	*Refine* 1
InitialStep auto	*FixedStep* auto	*MaxStep* auto

Tabelle 1.2. Valve_ Summary Information

NumModelInputs	N/A	*NumModelOutputs*	N/A
NumVirtualSubsystems	N/A	*NumNonvirtSubsystems*	N/A
NumNonVirtBlocksInModel	N/A	*NumBlockTypeCounts*	N/A
NumBlockSignals	N/A	*NumBlockParams*	N/A
NumZCEvents	N/A	*NumNonsampledZCs*	N/A

Systems

Name	**Parent**	**Snapshot**	**Blocks**	**Signals**
Valve_	<root>		Valve_	Valve_<1> Valve_<2>

Blocks

Tabelle 1.3. Block Type Count

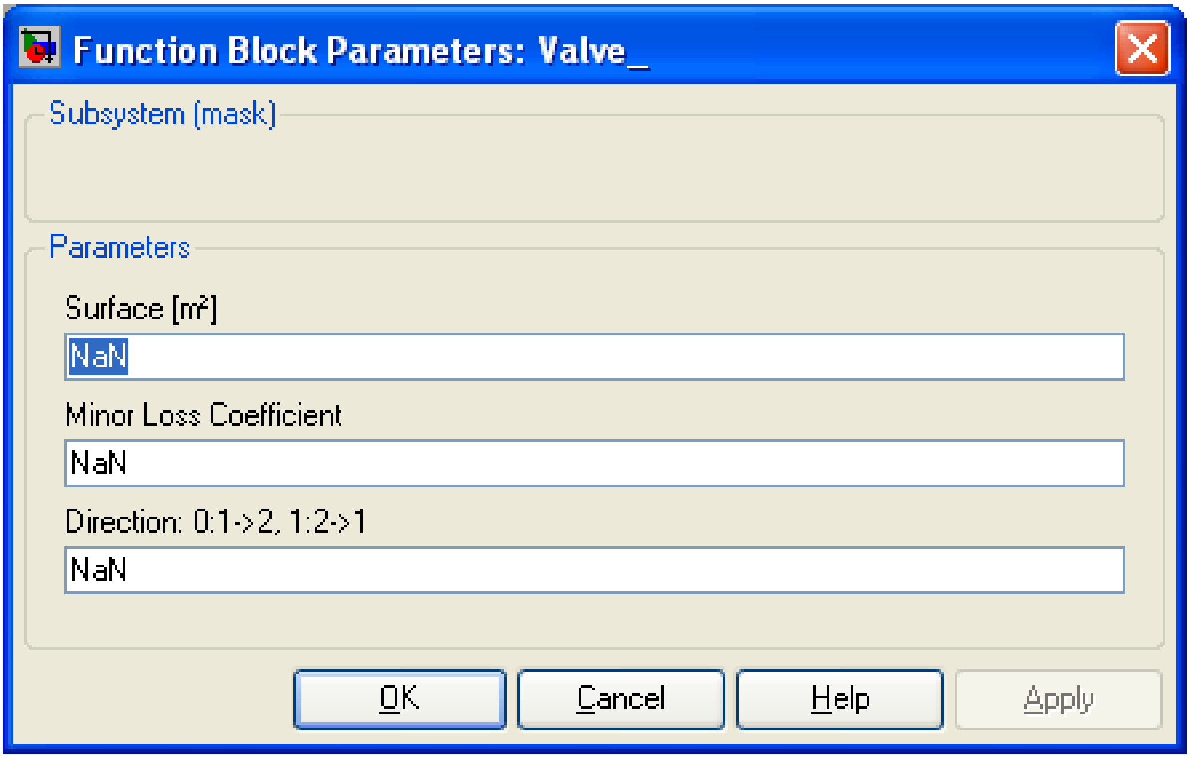
BlockType	**Count**	**Block Names**
Inport	19	In_1, In_2, beta, A, beta, zeta, p_in_1, rho_in_1, T_in_1, x_H2O_gas_in_1, x_CO2_in_1, x_H2O_liq_in_1, p_in_2, rho_in_2, T_in_2, x_H2O_gas_in_2, x_CO2_in_2, x_H2O_liq_in_2, Dir
Outport	11	rho, T, x_H2O_gas, x_CO2, x_H2O_liq, p_1, m_dot_air_1, p_2, m_dot_air_2, Out_1, Out_2
Constant	4	A, Dir, m_air, zeta

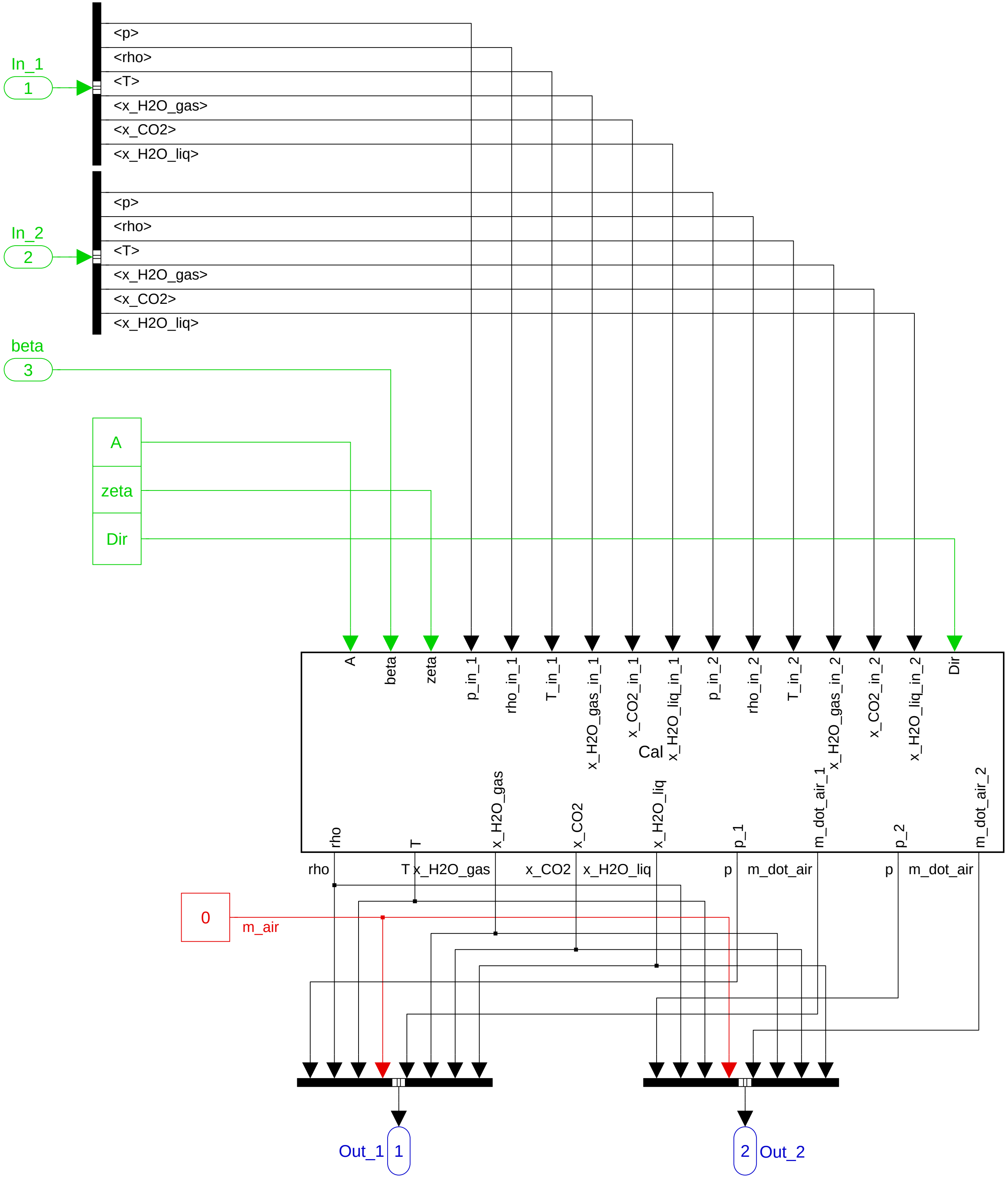
BlockType	Count	Block Names
BusSelector	2	Bus Selector1, Bus Selector3
BusCreator	2	Bus Creator1, Bus Creator2
Terminator	1	Terminator
SubSystem	1	Valve_
Stateflow (m)	1	Embedded MATLA Function
S-Function	1	SFunction
Demux	1	Demux

Data and Functions

Tabelle 1.4. Model Functions

Function Name	Parent Blocks	Calling string
NaN	 Valve_ Valve_ Valve_	 `NaN` `NaN` `NaN`


Function Block Parameters: Valve_
Subsystem (mask)
Parameters
Surface [m²]
NaN
Minor Loss Coefficient
NaN
Direction: 0:1->2, 1:2->1
NaN
OK
Cancel
Help
Apply


In_1
1
<p>
<rho>
<T>
<x_H2O_gas>
<x_CO2>
<x_H2O_liq>
In_2
2
<p>
<rho>
<T>
<x_H2O_gas>
<x_CO2>
<x_H2O_liq>
beta
3
A
zeta
Dir
A
beta
zeta
p_in_1
rho_in_1
T_in_1
x_H2O_gas_in_1
x_CO2_in_1
x_H2O_liq_in_1
p_in_2
rho_in_2
T_in_2
x_H2O_gas_in_2
x_CO2_in_2
x_H2O_liq_in_2
Dir
Cal
rho
T
x_H2O_gas
x_CO2
x_H2O_liq
p_1
m_dot_air_1
p_2
m_dot_air_2
rho
T
x_H2O_gas
x_CO2
x_H2O_liq
p
m_dot_air
p
m_dot_air
0
m_air
Out_1
1
2
Out_2

```
function [rho,T,x_H2O_gas,x_CO2,x_H2O_liq,p_1,m_dot_air_1,p_2,m_dot_air_2]=Cal(A,beta,↙
zeta,p_in_1,rho_in_1,T_in_1,x_H2O_gas_in_1,x_CO2_in_1,x_H2O_liq_in_1,p_in_2,rho_in_2,↙
T_in_2,x_H2O_gas_in_2,x_CO2_in_2,x_H2O_liq_in_2,Dir)

% ************************************************************************
% * Definition of a Valve (incompressible)
% *
% * Number of inputs:                  2
% *
% * Parameter: Surface:                A
% *            Opening factor :        OF
% *            Minor loss coefficient: zeta
% *            Flow Direction:         Dir
% *
% *
% * Relevant input variables of Valve
% *
% * Pressure:                          p_in
% * Density:                           rho_in
% * Temperature:                       T_in
% * Content water vapor:               x_H2O_gas_in
% * Content CO2:                       x_CO2_in
% * Content water:                     x_H2O_liq_in
% *
% *
% * Relevant output variables of Valve
% *
% * Temperature:                       T
% * Mass flow dry air:                 m_dot_air
% * Content water vapor:               x_H2O_gas
% * Content CO2:                       x_CO2
% * Content water:                     x_H2O_liq
% *
% ***********************************************************************
% * Embedded Matlab Function Cal:
% *
% * Calculations:
% * 1. Calculation parameter.
% * 2. Calculation of the state variables.
% * 3. Calculation flow velocity.
% * 4. Calculation mass flow.
% *
% *
% * Assumptions:
% * 2. State variables = input variables higher pressure
% * 3. The overall pressure drop is transferred into kinetic energy.
% * 4. The mass flow is calculated with a incompressible mass flow equation.
% *    This equation is only applicable for systems with low flow velocities.
% *
% *
% * Last modification : 15.03.2008
% * Author : Christian Müller(HAW)
% *
% ***********************************************************************

% * 1. Calculation parameter
```

```
A_eff          = sin(pi*beta/180)*A;

if Dir == 0
   if p_in_2 > p_in_1
      A_eff    = 0;
   end
end
if Dir == 1
   if p_in_1 > p_in_2
      A_eff    = 0;
   end
end
% *************************************************************************

% * 2. Calculation of the state variables.
rho            = 0;
T              = 0;
x_H2O_gas      = 0;
x_CO2          = 0;
x_H2O_liq      = 0;

if p_in_1 >= p_in_2
   rho         = rho_in_1;
   T           = T_in_1;
   x_H2O_gas   = x_H2O_gas_in_1;
   x_CO2       = x_CO2_in_1;
   x_H2O_liq   = x_H2O_liq_in_1;
else
   rho         = rho_in_2;
   T           = T_in_2;
   x_H2O_gas   = x_H2O_gas_in_2;
   x_CO2       = x_CO2_in_2;
   x_H2O_liq   = x_H2O_liq_in_2;
end
% *************************************************************************

% * 3. Calculation flow velocity
p_1            = p_in_2;
p_2            = p_in_1;
Delta_p        = p_in_1-p_in_2;
rho_air        = rho/(1+x_H2O_gas+x_CO2);

if abs(Delta_p) < 1
   Delta_p     = 0;
end

zeta_total     = 1+zeta;
v              = sqrt(abs(Delta_p)*(2/rho_air)*(1/zeta_total));
% *************************************************************************

% * 4. Calculation mass flow
m_dot_air      = A_eff*rho_air*v;

if Delta_p >= 0
   m_dot_air_1 = -m_dot_air;
   m_dot_air_2 =  m_dot_air;
```

```
else
   m_dot_air_1 =  m_dot_air;
   m_dot_air_2 = -m_dot_air;
end
% ***********************************************************************
```