Simulink Report: Specialized_Fan_

Christian Müller

2008-03-06

Model - Specialized_Fan_

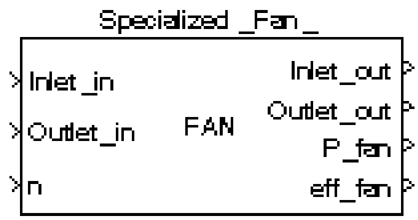


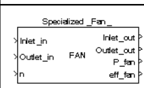
Tabelle 1.1. Specialized_Fan_ Simulation Parameters

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. Specialized_Fan_ 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**
Specialized_ Fan_	<root>		Specialized_Fan_	Specialized_Fan_<1> Specialized_Fan_<2> Specialized_Fan_<3> Specialized_Fan_<4>

Blocks

Tabelle 1.3. Block Type Count

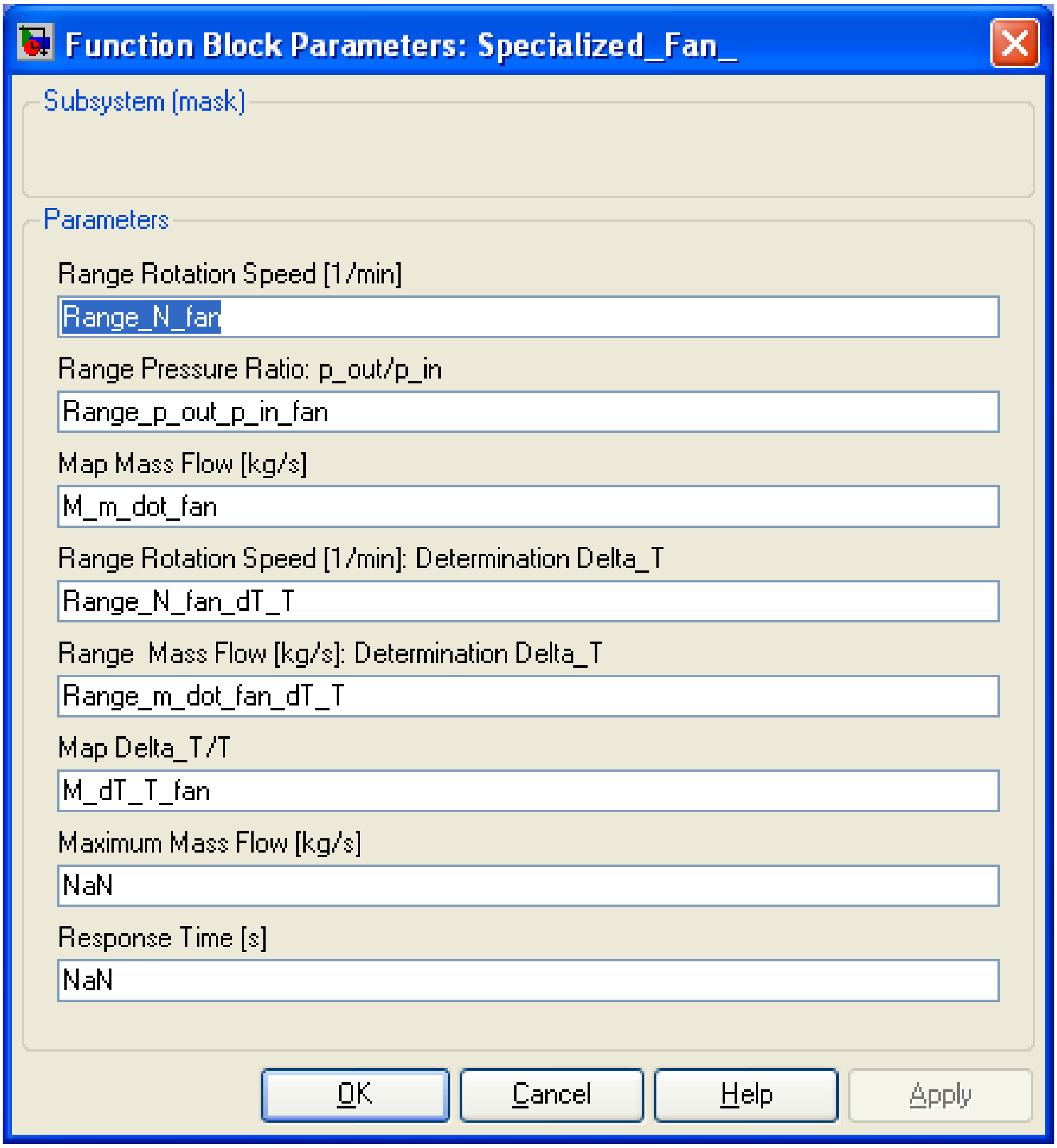
BlockType	**Count**	**Block Names**
Inport	19	Inlet_in, Outlet_in, n, p_0, T_0, p_inlet, p_outlet, T_inlet, x_H2O_gas, x_CO2, dT_T, m_dot_air_corr, T_inlet, x_H2O_gas, x_CO2, T_air_inlet, T_0, n, In
Outport	13	rho_outlet, T_outlet, m_dot_air_inlet, m_dot_air_outlet, P_fan, eff, T_air_inlet, N, Out, Cold_inlet_out, Cold_outlet_out, P_comp, eff

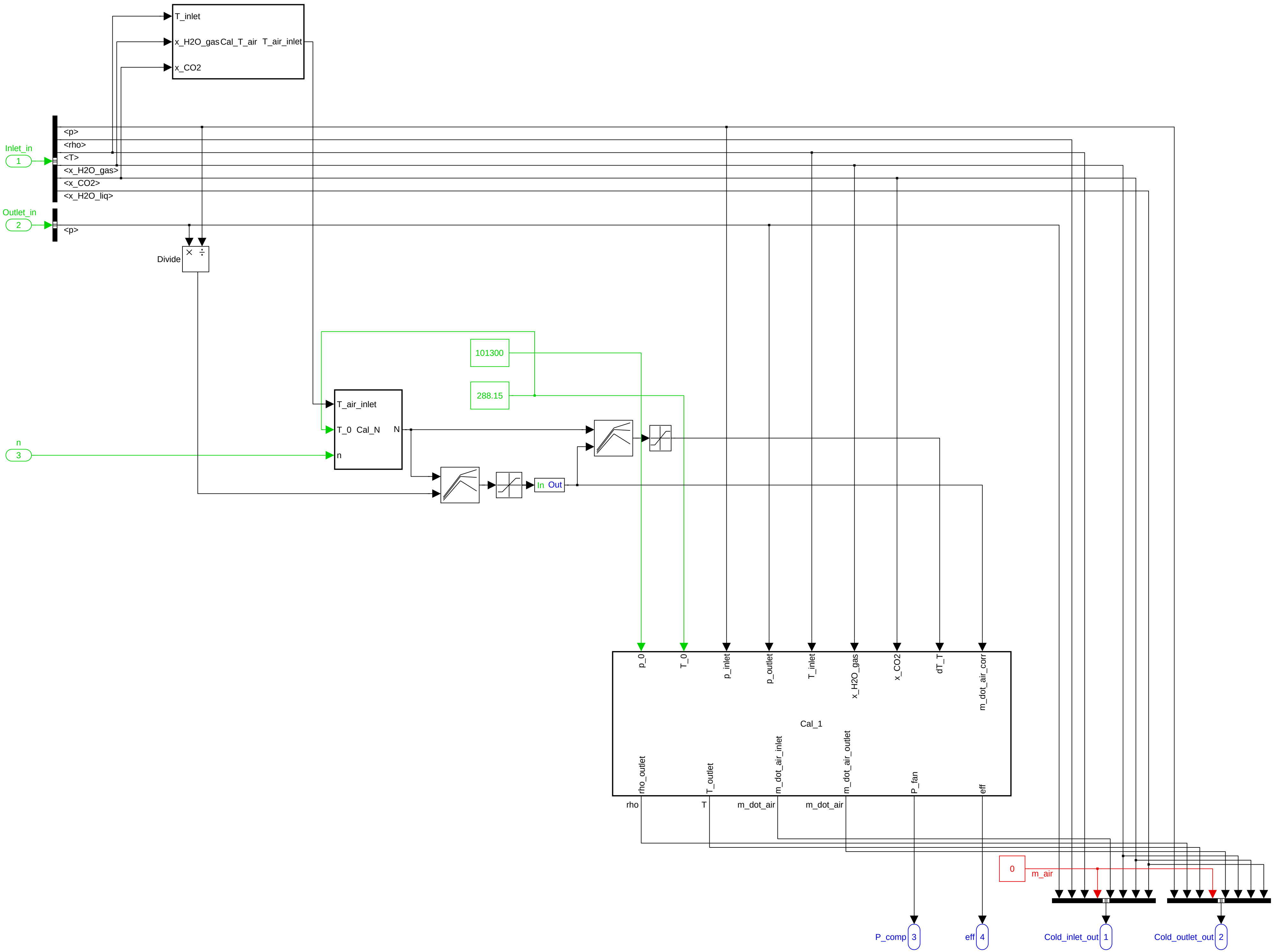
BlockType	Count	Block Names
Terminator	3	Terminator , Terminator , Terminator
Stateflow (m)	3	Embedded MATLA Function1, Embedded MATLA Function2, Embedded MATLA Function3
S-Function	3	SFunction , SFunction , SFunction
Demux	3	Demux , Demux , Demux
Constant	3	T_0, m_air, p_0
SubSystem	2	Specialized_Fan_, Subsystem11
Saturate	2	Saturation, Saturation1
Lookup2D	2	Lookup Table (2-D), Lookup Table (2-D)1
BusSelector	2	Bus Selector, Bus Selector3
BusCreator	2	Bus Creator1, Bus Creator2
TransferFcn	1	Transfer Fcn
Product	1	Divide

Data and Functions

Tabelle 1.4. Model Functions

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


Function Block Parameters: Specialized_Fan_
Subsystem (mask)
Parameters
Range Rotation Speed [1/min]
Range_N_fan
Range Pressure Ratio: p_out/p_in
Range_p_out_p_in_fan
Map Mass Flow [kg/s]
M_m_dot_fan
Range Rotation Speed [1/min]: Determination Delta_T
Range_N_fan_dT_T
Range Mass Flow [kg/s]: Determination Delta_T
Range_m_dot_fan_dT_T
Map Delta_T/T
M_dT_T_fan
Maximum Mass Flow [kg/s]
NaN
Response Time [s]
NaN
OK
Cancel
Help
Apply



T_inlet
x_H2O_gas Cal_T_air T_air_inlet
x_CO2
Inlet_in
1
<p>
<rho>
<T>
<x_H2O_gas>
<x_CO2>
<x_H2O_liq>
Outlet_in
2
<p>
Divide
101300
288.15
T_air_inlet
T_0 Cal_N N
n
n
3
In Out
p_0
T_0
p_inlet
p_outlet
T_inlet
x_H2O_gas
x_CO2
dT_T
m_dot_air_corr
Cal_1
rho_outlet
T_outlet
m_dot_air_inlet
m_dot_air_outlet
P_fan
eff
rho
T
m_dot_air
m_dot_air
0
m_air
P_comp 3
eff 4
Cold_inlet_out 1
Cold_outlet_out 2

```
function T_air_inlet = Cal_T_air(T_inlet,x_H2O_gas,x_CO2)

% ************************************************************************
% * Definition of a specialized fan
% *
% * Number of inputs :                4
% *
% * Parameter: Characteristic Maps:    Mass Flow, Efficiency
% *
% *
% * Relevant input variables of Specialized_Fan
% *
% * 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 Specialized_Fan
% *
% * 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_T_air:
% *
% * Calculations:
% * 1. Calculation dry bulb temperature T_air.
% *
% *
% * Assumptions:
% * 1. The specific enthalpy of the inflowing gas mixture is equal the enthalpy
% *    of a dry air flow.
% *
% *
% * Last modification : 15.03.2008
% * Author : Christian Müller(HAW)
% *
% ***********************************************************************

% * 1. Calculation dry bulb temperature T_air
c_p_air     = 1005;
c_p_H2O_gas = 1870;
c_p_CO2     = 830;
r_0         = 2500000;
T_air_inlet =↙
(c_p_air*T_inlet+x_H2O_gas*c_p_H2O_gas*T_inlet+x_CO2*c_p_CO2*T_inlet+x_H2O_gas*r_0)↙
/c_p_air;
% ***********************************************************************
```

```
function N = Cal_N(T_air_inlet,T_0,n)

% *************************************************************************
% * Definition of a specialized fan
% *
% * Number of inputs :                 4
% *
% * Parameter: Characteristic Maps:    Mass Flow, Efficiency
% *
% *
% * Relevant input variables of Specialized_Fan
% *
% * 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 Specialized_Fan
% *
% * 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_N:
% *
% * Calculations:
% * 1. Calculation corrected rotational speed N.
% *
% *
% * Last modification : 15.03.2008
% * Author : Christian Müller(HAW)
% *
% ************************************************************************

% * 1. Calculation corrected rotational speed N
N = n/sqrt(T_air_inlet/T_0);
% ************************************************************************
```

```
function [rho_outlet,T_outlet,m_dot_air_inlet,m_dot_air_outlet,P_fan,eff] = Cal_1(p_0,↙
T_0,p_inlet,p_outlet,T_inlet,x_H2O_gas,x_CO2,dT_T,m_dot_air_corr)

% *************************************************************************
% * Definition of a specialized fan
% *
% * Number of inputs :                  4
% *
% * Parameter: Characteristic Maps:     Mass Flow, Efficiency
% *
% *
% * Relevant input variables of Specialized_Fan
% *
% * 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 Specialized_Fan
% *
% * 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_1:
% *
% * Calculations:
% * 1. Definition specific gas constants.
% * 2. Calculation mass flow.
% * 3. Calculation absorbed power and output variables.
% *
% *
% * Last modification : 15.03.2008
% * Author : Christian Müller(HAW)
% *
% ************************************************************************

% * 1. Definition specific gas constants
R_air             = 287.058;
R_H2O_gas         = 461.523;
R_CO2             = 188.924;
c_p_air           = 1005;
c_p_H2O_gas       = 1870;
c_p_CO2           = 830;

R_avg             = (R_air+x_H2O_gas*R_H2O_gas+x_CO2*R_CO2)/(1+x_H2O_gas+x_CO2);
c_p_avg           = (c_p_air+x_H2O_gas*c_p_H2O_gas+x_CO2*c_p_CO2)/(1+x_H2O_gas+x_CO2);
c_v_avg           = c_p_avg-R_avg;
gamma_avg         = c_p_avg/c_v_avg;
% ************************************************************************
```

```
% * 2. Calculation mass flow
m_dot_air          = m_dot_air_corr*(p_inlet/p_0)*(1/sqrt(T_inlet/T_0));
% ************************************************************************

% * 3. Calculation absorbed power and output variables
eff                = 0;

if abs(dT_T) > 0
   eff             = (((p_outlet/p_inlet)^((gamma_avg-1)/gamma_avg))-1)/dT_T;
   if eff > 1
      eff          = 1;
   end
end

P_fan              = 0;
T_outlet           = T_inlet;

if eff > 0.01
   P_fan           = (1/eff)*m_dot_air*c_p_air*T_inlet*(((p_outlet/p_inlet)^↙
((gamma_avg-1)/gamma_avg))-1);

   if p_outlet < p_inlet
      P_fan        = 0;
   end

   T_outlet        = T_inlet*(1+dT_T);
end

if P_fan < 0
   P_fan           = 0;
end

rho_outlet         = p_outlet/(R_avg*T_outlet);
m_dot_air_inlet    = -m_dot_air;
m_dot_air_outlet   =  m_dot_air;

if m_dot_air < 0
   m_dot_air_inlet = 0;
end
% ************************************************************************
```