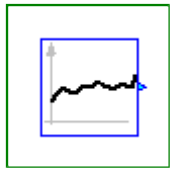
First_Thermo.ThermoSysPro.Experiments

contains the components with failures

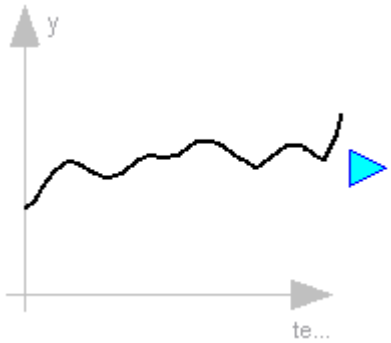
Package Content

Name	Description
Expression	varying expression
Pipe3	Lumped straight pipe (circular duct) with failures
SwitchValve	Switch valve with failures
SwitchValve7	Switch valve with failures
SwitchValve8	Switch valve with failures
Pump2	StaticCentrifugalPump with failures
PumpN15	CentrifugalPump with failures

First_Thermo.ThermoSysPro.Experiments.Expression



varying expression



Information

Adapted from the Modelica.Blocks.Sources librarys

Version 1.0

Parameters

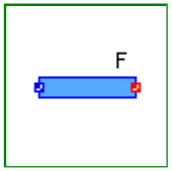
Type	Name	Default	Description
Real	exp	1	equation of the output (time-varying)

Connectors

Type	Name	Description
OutputReal	y	output

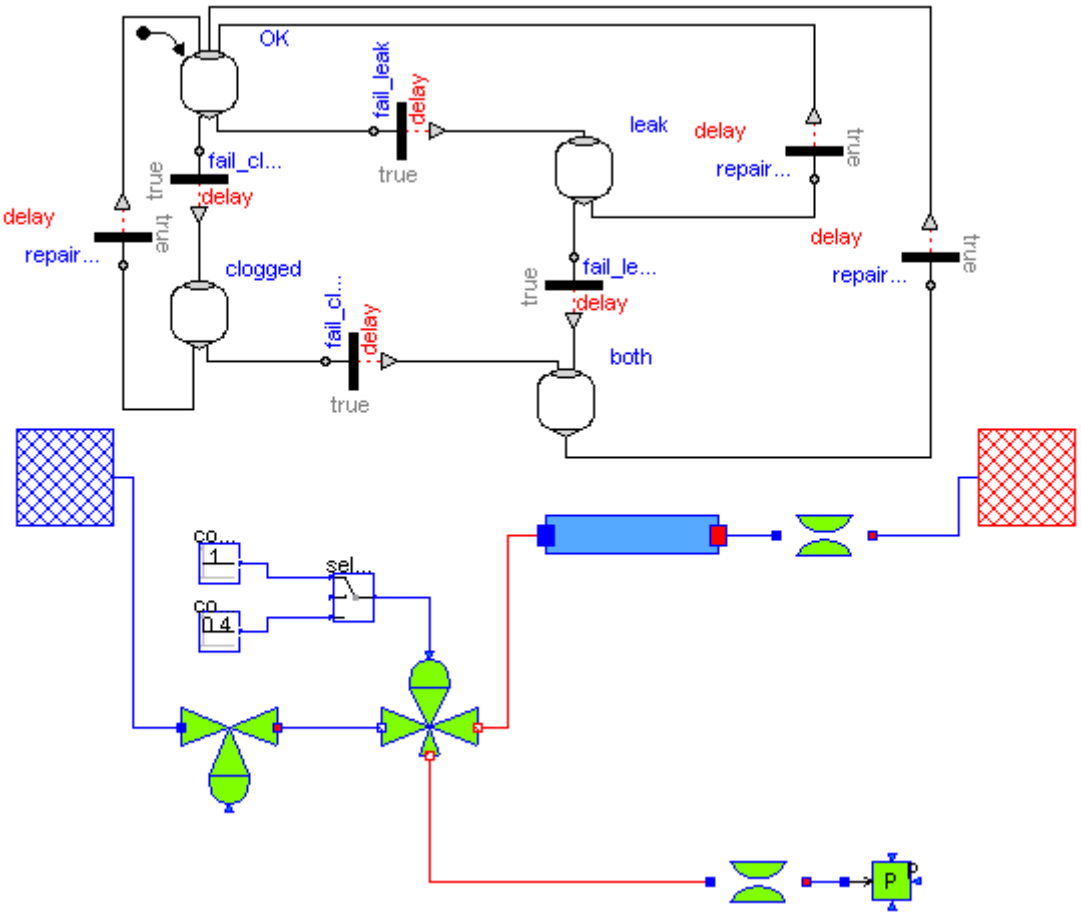
Modelica definition

```
block Expression "varying expression"
  Real exp=1 "equation of the output (time-varying)";
  ThermoSysPro.InstrumentationAndControl.Connectors.OutputReal y "output";
equation

  y.signal = exp;
end Expression;
```



First_Thermo.ThermoSysPro.Experiments.Pipe3

Lumped straight pipe (circular duct) with failures



Information

The pipe can present a leak, a partial clogging, or both.

Parameters

Type	Name	Default	Description
Length	L	10.	Pipe length [m]
Diameter	D	0.2	Pipe internal hydraulic diameter [m]
Real	lambda	0.03	Friction pressure loss coefficient (active if lambda_fixed=true)
Real	rugosrel	0.0001	Pipe roughness (active if lambda_fixed=false)
Position	z1	0	Inlet altitude [m]
Position	z2	0	Outlet altitude [m]
Boolean	lambda_fixed	true	true: lambda given by parameter - false: lambde computed using Idel'Cik correlation
Boolean	inertia	false	true: momentum balance equation with inertia - false: without inertia
Boolean	continuous_flow_reversal	false	true: continuous flow reversal - false: discontinuous flow reversal
Integer	fluid	1	1: water/steam - 2: C3H3F5
Density	p_rho	0	If > 0, fixed fluid density [kg/m3]
Integer	mode	0	IF97 region. 1:liquid - 2:steam - 4:saturation line - 0:automatic

Connectors

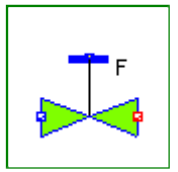
Type	Name	Description
FluidInletI	fluidInletI	
FluidOutletI	fluidOutletI	

Modelica definition

```
model Pipe3 "Lumped straight pipe (circular duct) with failures"

  parameter Modelica.SIunits.Length L=10. "Pipe length";
  parameter Modelica.SIunits.Diameter D=0.2 "Pipe internal hydraulic diameter";
  parameter Real lambda=0.03     "Friction pressure loss coefficient (active if lambda_fixed=true)";
  parameter Real rugosrel=0.0001     "Pipe roughness (active if lambda_fixed=false)";
  parameter Modelica.SIunits.Position z1=0 "Inlet altitude";
  parameter Modelica.SIunits.Position z2=0 "Outlet altitude";
  parameter Boolean lambda_fixed=true     "true: lambda given by parameter - false: lambde computed using Idel'Cik correlation";
  parameter Boolean inertia=false     "true: momentum balance equation with inertia - false: without inertia";
  parameter Boolean continuous_flow_reversal=false     "true: continuous flow reversal - false: discontinuous flow reversal";
  parameter Integer fluid=1 "1: water/steam - 2: C3H3F5";
  parameter Modelica.SIunits.Density p_rho=0 "If > 0, fixed fluid density";
  parameter Integer mode=0     "IF97 region. 1:liquid - 2:steam - 4:saturation line - 0:automatic";
```

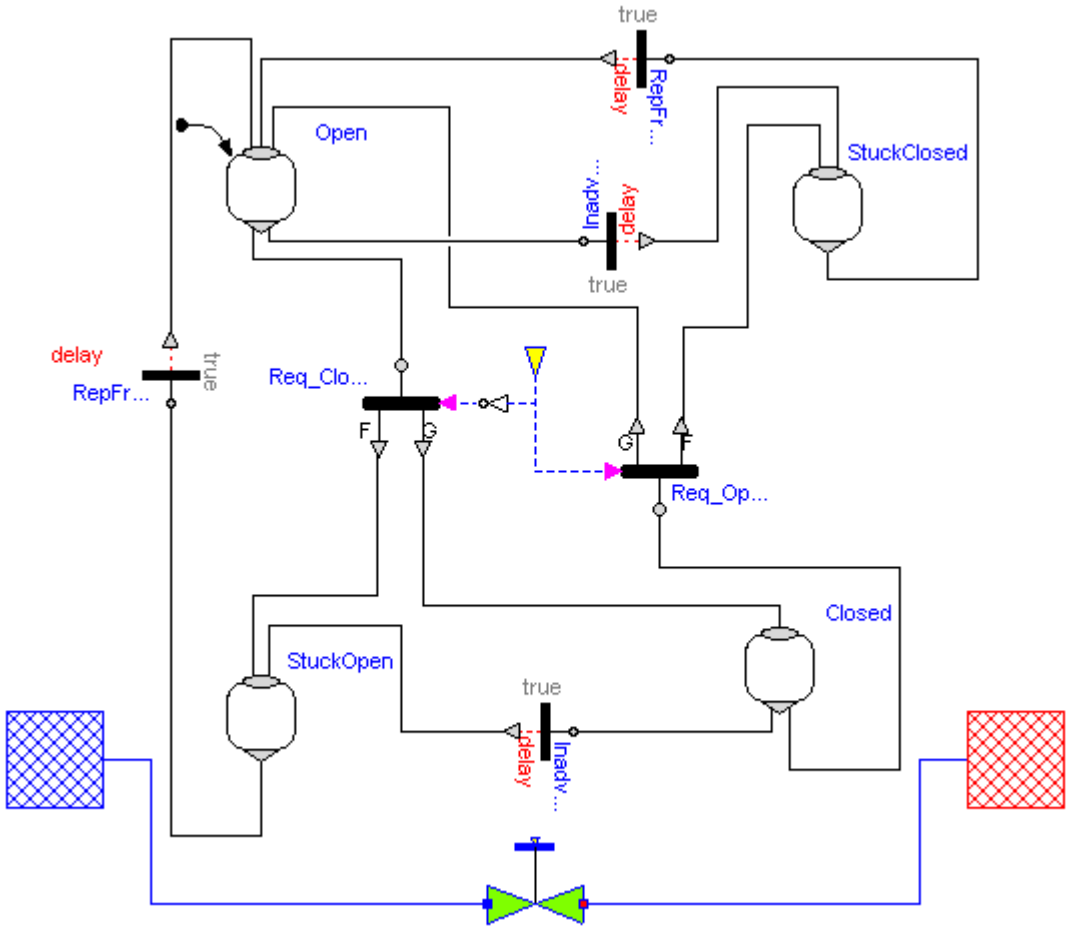
```
WaterSteam.PressureLosses.ThreeWayValve     splitter2_1     "control of the leaking (1 = no leak)";
WaterSteam.BoundaryConditions.SinkP             sink1 "sink for the leak";
WaterSteam.PressureLosses.SingularPressureLoss
  singularPressureLoss;
WaterSteam.PressureLosses.SingularPressureLoss
  singularPressureLoss1;
WaterSteam.PressureLosses.LumpedStraightPipe              lumpedStraightPipe(
  L=L,
  D=D,
  lambda=lambda,
  rugosrel=rugosrel,
  z1=z1,
  z2=z2,
  lambda_fixed=lambda_fixed,
  inertia=inertia,
  continuous_flow_reversal=continuous_flow_reversal,
  fluid=fluid,
  p_rho=p_rho,
  mode=mode) "standard pipe";
InstrumentationAndControl.Blocks.NonLineaire.Selecteur             selecteur;
InstrumentationAndControl.Blocks.Sources.Constante             constante(k=1);
InstrumentationAndControl.Blocks.Sources.Constante             constante1(k=0.4);
Modelica_StateGraph2.Step OK(
  nOut=2,
  nIn=3,
  initialStep=true) "normal pipe";
Modelica_StateGraph2.Step clogged(nIn=1, nOut=2) "partially clogged pipe";
Transitions_detailed.newTransitions.MultiTransition fail_clogged(immediate=false,
    kind=2);
Transitions_detailed.newTransitions.MultiTransition repair_clog2OK(
  immediate=false,
  kind=1,
  delayTime=0.5);
WaterSteam.Connectors.FluidInletI
                      fluidInletI;
WaterSteam.Connectors.FluidOutletI
                       fluidOutletI;
WaterSteam.PressureLosses.ControlValve controlValve     "control of the clogging (0 = completely clogged)";
Modelica_StateGraph2.Step leak(nIn=1, nOut=2) "leaking pipe";
Transitions_detailed.newTransitions.MultiTransition fail_leak(immediate=false,
    kind=2);
Transitions_detailed.newTransitions.MultiTransition repair_clog2OK1(
  immediate=false,
  kind=1,
  delayTime=0.5);
Modelica_StateGraph2.Step both(nOut=1, nIn=2)     "partially clogged and leaking pipe";
Transitions_detailed.newTransitions.MultiTransition fail_clog2both(immediate=false,
```

```
      kind=2);
  Transitions_detailed.newTransitions.MultiTransition fail_leak2both(immediate=false,
      kind=2);
  Transitions_detailed.newTransitions.MultiTransition repair_both2OK(
    immediate=false,
    kind=1,
    delayTime=0.5);
equation
  // control of the leak
  selecteur.uCond.signal = OK.active or clogged.active;
  // control of the clogging
  controlValve.Ouv.signal = if OK.active or leak.active then 1 else 0.6;
  connect(singularPressureLoss1.C2,sink1. C);
  connect(lumpedStraightPipe.C2,singularPressureLoss. C1);
  connect(constante1.y,selecteur. u2);
  connect(constante.y,selecteur. u1);
  connect(clogged.inPort[1], fail_clogged.outPort);
  connect(fail_clogged.inPort, OK.outPort[1]);
  connect(repair_clog2OK.inPort, clogged.outPort[1]);
  connect(repair_clog2OK.outPort, OK.inPort[1]);
  connect(singularPressureLoss.C2, fluidOutletI);
  connect(controlValve.C1, fluidInletI);
  connect(fail_leak.inPort, OK.outPort[2]);
  connect(leak.inPort[1], fail_leak.outPort);
  connect(repair_clog2OK1.outPort, OK.inPort[3]);
  connect(fail_clog2both.inPort, clogged.outPort[2]);
  connect(fail_leak2both.inPort, leak.outPort[1]);
  connect(repair_clog2OK1.inPort, leak.outPort[2]);
  connect(fail_clog2both.outPort, both.inPort[1]);
  connect(fail_leak2both.outPort, both.inPort[2]);
  connect(repair_both2OK.inPort, both.outPort[1]);
  connect(repair_both2OK.outPort, OK.inPort[2]);
  connect(controlValve.C2, splitter2_1.C1);
  connect(splitter2_1.Ouv, selecteur.y);
  connect(splitter2_1.C2, lumpedStraightPipe.C1);
  connect(splitter2_1.C3, singularPressureLoss1.C1);
end Pipe3;
```



First_Thermo.ThermoSysPro.Experiments.SwitchValve

Switch valve with failures



Information

This valve can present failures only linked to its open or closed states.

Parameters

Type	Name	Default	Description
PressureLossCoefficient	k	1000	Pressure loss coefficient [m-4]
MassFlowRate	Qmin	1.e-6	Mass flow when the valve is closed [kg/s]
Boolean	continuous_flow_reversal	false	true: continuous flow reversal - false: discontinuous flow reversal
Integer	fluid	1	1: water/steam - 2: C3H3F5
Density	p_rho	0	If > 0, fixed fluid density [kg/m3]
Integer	mode	0	IF97 region. 1:liquid - 2:steam - 4:saturation line - 0:automatic

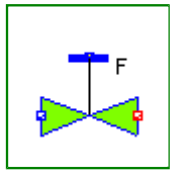
Connectors

Type	Name	Description
FluidInletI	fluidInletI	
FluidOutletI	fluidOutletI	
InputLogical	requestOpening	requested opening of the valve : true --> open

Modelica definition

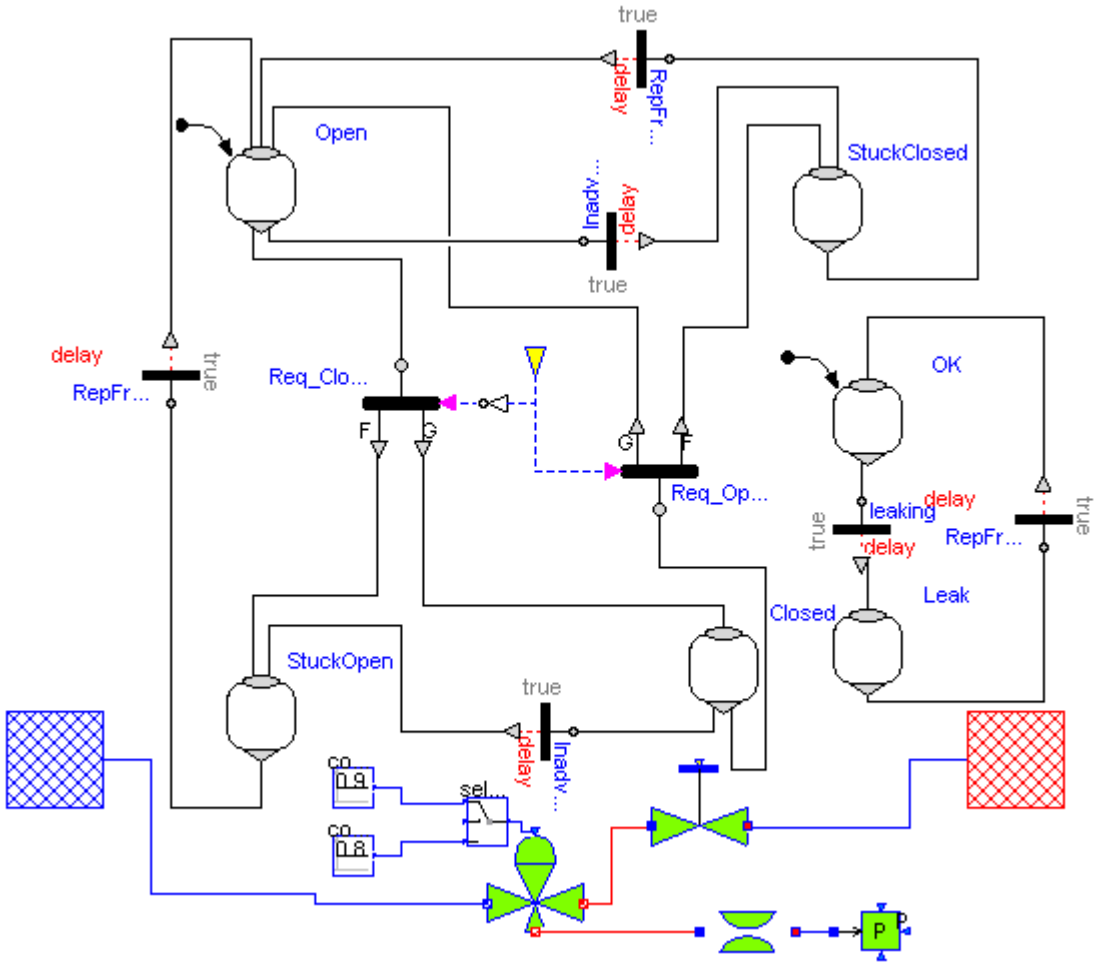
```
model SwitchValve "Switch valve with failures"
  parameter ThermoSysPro.Units.PressureLossCoefficient k=1000     "Pressure loss coefficient";
  parameter Modelica.SIunits.MassFlowRate Qmin=1.e-6     "Mass flow when the valve is closed";
  parameter Boolean continuous_flow_reversal=false     "true: continuous flow reversal - false: discontinuous flow reversal";
  parameter Integer fluid=1 "1: water/steam - 2: C3H3F5";
  parameter Modelica.SIunits.Density p_rho=0 "If > 0, fixed fluid density";
  parameter Integer mode=0     "IF97 region. 1:liquid - 2:steam - 4:saturation line - 0:automatic";

  WaterSteam.Connectors.FluidInletI fluidInletI;
  WaterSteam.Connectors.FluidOutletI fluidOutletI;
  WaterSteam.PressureLosses.SwitchValve switchValve "standard switch valve";
  Modelica_StateGraph2.Step Open(
    nOut=2,
    nIn=3,
    initialStep=true) "open valve";
  Transitions_detailed.newTransitions.MultiTransition Inadv_Opening(immediate=false,
      kind=2) "inadvertent opening of the valve";
  Modelica_StateGraph2.Step StuckClosed(nIn=2, nOut=1) "stuck closed valve";
  Modelica_StateGraph2.Step StuckOpen(nIn=2, nOut=1) "stuck open valve";
  Modelica_StateGraph2.Step Closed(nIn=1, nOut=2) "closed valve";
  Transitions_detailed.newTransitions.TransitionOnRequest Req_Closing     "requested closing of the valve";
  Transitions_detailed.newTransitions.TransitionOnRequest Req_Opening     "requested opening of the valve";
```

```
  Transitions_detailed.newTransitions.MultiTransition Inadv_Closing(immediate=false,
      kind=2) "inadvertent closing of the valve";
  Transitions_detailed.newTransitions.MultiTransition RepFromClosed(immediate=false,
      kind=3) "reparation from stuck closed state to open state";
  Transitions_detailed.newTransitions.MultiTransition RepFromOpen(immediate=false,
      kind=3) "reparation from stuck open state to open state";
  InstrumentationAndControl.Connectors.InputLogical requestOpening     "requested opening of the valve : true --> open";
equation
  switchValve.Ouv.signal = Open.active or StuckOpen.active;
  Req_Opening.request = requestOpening.signal;
  Req_Closing.request = not requestOpening.signal;
  connect(switchValve.C1, fluidInletI);
  connect(Open.outPort[1], Req_Closing.inPort);
  connect(Req_Closing.goodPath, Closed.inPort[1]);
  connect(Req_Closing.failingPath, StuckOpen.inPort[1]);
  connect(Closed.outPort[2], Req_Opening.inPort);
  connect(Req_Opening.goodPath, Open.inPort[3]);

  connect(Req_Opening.failingPath, StuckClosed.inPort[1]);
  connect(Inadv_Closing.inPort, Open.outPort[2]);
  connect(Inadv_Closing.outPort, StuckClosed.inPort[2]);
  connect(StuckClosed.outPort[1], RepFromClosed.inPort);
  connect(RepFromClosed.outPort, Open.inPort[2]);
  connect(Inadv_Opening.inPort, Closed.outPort[1]);
  connect(Inadv_Opening.outPort, StuckOpen.inPort[2]);
  connect(StuckOpen.outPort[1], RepFromOpen.inPort);
  connect(RepFromOpen.outPort, Open.inPort[1]);
  connect(switchValve.C2, fluidOutletI);
end SwitchValve;
```



First_Thermo.ThermoSysPro.Experiments.SwitchValve7

Switch valve with failures



Information

This valve can present failures linked to its open or closed states, and can present a leak on its left (blue) side.

Parameters

Type	Name	Default	Description
PressureLossCoefficient	k	1000	Pressure loss coefficient [m-4]
MassFlowRate	Qmin	1.e-6	Mass flow when the valve is closed [kg/s]
Boolean	continuous_flow_reversal	false	true: continuous flow reversal - false: discontinuous flow reversal
Integer	fluid	1	1: water/steam - 2: C3H3F5
Density	p_rho	0	If > 0, fixed fluid density [kg/m3]
Integer	mode	0	IF97 region. 1:liquid - 2:steam - 4:saturation line - 0:automatic

Connectors

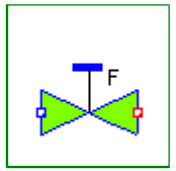
Type	Name	Description
FluidInletI	fluidInletI	
FluidOutletI	fluidOutletI	
InputLogical	requestOpening	requested opening of the valve : true --> open

Modelica definition

```
model SwitchValve7 "Switch valve with failures"
  parameter ThermoSysPro.Units.PressureLossCoefficient k=1000     "Pressure loss coefficient";
  parameter Modelica.SIunits.MassFlowRate Qmin=1.e-6     "Mass flow when the valve is closed";
  parameter Boolean continuous_flow_reversal=false     "true: continuous flow reversal - false: discontinuous flow reversal";
  parameter Integer fluid=1 "1: water/steam - 2: C3H3F5";
  parameter Modelica.SIunits.Density p_rho=0 "If > 0, fixed fluid density";
  parameter Integer mode=0     "IF97 region. 1:liquid - 2:steam - 4:saturation line - 0:automatic";

  WaterSteam.Connectors.FluidInletI fluidInletI;
  WaterSteam.Connectors.FluidOutletI fluidOutletI;
  WaterSteam.PressureLosses.SwitchValve switchValve "standard switch valve";
  Modelica_StateGraph2.Step Open(
    nOut=2,
    nIn=3,
    initialStep=true) "open valve";
  Transitions_detailed.newTransitions.MultiTransition Inadv_Opening(immediate=false,
      kind=2) "inadvertent opening of the valve";
  Modelica_StateGraph2.Step StuckClosed(nIn=2, nOut=1) "stuck closed valve";
  Modelica_StateGraph2.Step StuckOpen(nIn=2, nOut=1) "stuck open valve";
  Modelica_StateGraph2.Step Closed(nIn=1, nOut=2) "closed valve";
  Transitions_detailed.newTransitions.TransitionOnRequest Req_Closing     "requested closing of the valve";
  Transitions_detailed.newTransitions.TransitionOnRequest Req_Opening     "requested opening of the valve";
```

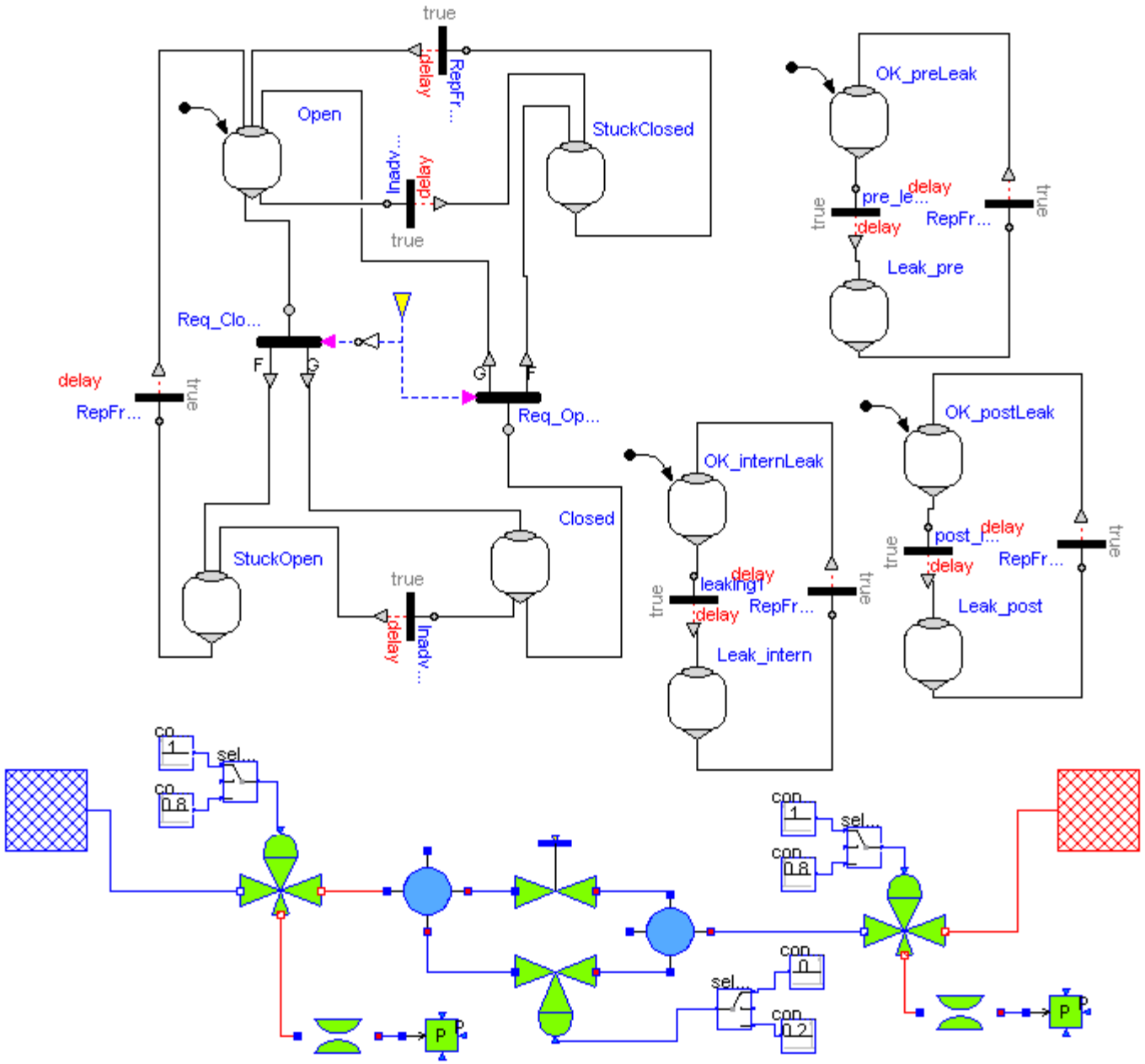
```
  Transitions_detailed.newTransitions.MultiTransition Inadv_Closing(immediate=false,
      kind=2) "inadvertent closing of the valve";
  Transitions_detailed.newTransitions.MultiTransition RepFromClosed(immediate=false,
      kind=3) "reparation from stuck closed state to open state";
  Transitions_detailed.newTransitions.MultiTransition RepFromOpen(immediate=false,
      kind=3) "reparation from stuck open state to open state";
  InstrumentationAndControl.Connectors.InputLogical requestOpening     "requested opening of the valve : true --> open";
  WaterSteam.PressureLosses.ThreeWayValve     splitter2_1     "control of the leaking (1 = no leak)";
  InstrumentationAndControl.Blocks.NonLineaire.Selecteur              selecteur;
  InstrumentationAndControl.Blocks.Sources.Constante              constante(k=0.9);
  InstrumentationAndControl.Blocks.Sources.Constante              constante1(k=0.8);
  WaterSteam.BoundaryConditions.SinkP             sink1 "sink for the leak";
  WaterSteam.PressureLosses.SingularPressureLoss
    singularPressureLoss1;
  Modelica_StateGraph2.Step Leak(nIn=1, nOut=1) "leaking closed valve";
  Transitions_detailed.newTransitions.MultiTransition leaking(immediate=false,
      kind=2) "inadvertent opening of the valve";
  Transitions_detailed.newTransitions.MultiTransition RepFromLeak(immediate=false,
      kind=3) "reparation from leaking state to OK state";
  Modelica_StateGraph2.Step OK(
    nOut=1,
    nIn=1,
    initialStep=true) "open valve";
equation
  switchValve.Ouv.signal = Open.active or StuckOpen.active;
  Req_Opening.request = requestOpening.signal;
  Req_Closing.request = not requestOpening.signal;
  selecteur.uCond.signal = not Leak.active;
  connect(Open.outPort[1], Req_Closing.inPort);
  connect(Req_Closing.goodPath, Closed.inPort[1]);
  connect(Req_Closing.failingPath, StuckOpen.inPort[1]);
  connect(Closed.outPort[2], Req_Opening.inPort);
  connect(Req_Opening.goodPath, Open.inPort[3]);

  connect(Req_Opening.failingPath, StuckClosed.inPort[1]);
  connect(Inadv_Closing.inPort, Open.outPort[2]);
  connect(Inadv_Closing.outPort, StuckClosed.inPort[2]);
  connect(StuckClosed.outPort[1], RepFromClosed.inPort);
  connect(RepFromClosed.outPort, Open.inPort[2]);
  connect(Inadv_Opening.inPort, Closed.outPort[1]);
  connect(Inadv_Opening.outPort, StuckOpen.inPort[2]);
  connect(StuckOpen.outPort[1], RepFromOpen.inPort);
  connect(RepFromOpen.outPort, Open.inPort[1]);
  connect(switchValve.C2, fluidOutletI);
  connect(constante1.y,selecteur. u2);
  connect(constante.y,selecteur. u1);
  connect(singularPressureLoss1.C2,sink1. C);
```

```
  connect(leaking.outPort, Leak.inPort[1]);
  connect(Leak.outPort[1], RepFromLeak.inPort);
  connect(leaking.inPort, OK.outPort[1]);
  connect(RepFromLeak.outPort, OK.inPort[1]);
  connect(selecteur.y, splitter2_1.Ouv);
  connect(splitter2_1.C1, fluidInletI);
  connect(splitter2_1.C3, singularPressureLoss1.C1);
  connect(splitter2_1.C2, switchValve.C1);
end SwitchValve7;
```



First_Thermo.ThermoSysPro.Experiments.**SwitchValve8**

Switch valve with failures


true
delay
RepFr...
Open
Inadv...
delay
true
StuckClosed
OK_preLeak
pre_le...
delay
true
delay
RepFr...
true
Leak_pre
Req_Clo...
F
G
G
F
delay
RepFr...
true
Req_Op...
OK_internLeak
OK_postLeak
Closed
StuckOpen
true
delay
Inadv...
leaking1
delay
true
delay
RepFr...
true
Leak_intern
post_...
delay
true
delay
RepFr...
true
Leak_post
con
1
sel...
con
0.8
con
1
sel...
con
0.8
con
n
sel...
con
0.2
P
P
P

Information

This valve can present failures linked to its open or closed states, a leak on the blue side, a leak on the red side, an an internal leak (can not be completely closed).

Parameters

Type	Name	Default	Description
PressureLossCoefficient	k	1000	Pressure loss coefficient [m-4]
MassFlowRate	Qmin	1.e-6	Mass flow when the valve is closed [kg/s]
Boolean	continuous_flow_reversal	false	true: continuous flow reversal - false: discontinuous flow reversal
Integer	fluid	1	1: water/steam - 2: C3H3F5
Density	p_rho	0	If > 0, fixed fluid density [kg/m3]
Integer	mode	0	IF97 region. 1:liquid - 2:steam - 4:saturation line - 0:automatic

Connectors

Type	Name	Description
FluidInletI	fluidInletI	
FluidOutletI	fluidOutletI	
InputLogical	requestOpening	requested opening of the valve : true -> open

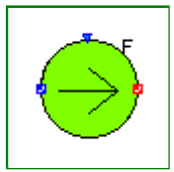
Modelica definition

```
model SwitchValve8 "Switch valve with failures"
  parameter ThermoSysPro.Units.PressureLossCoefficient k=1000     "Pressure loss coefficient";
  parameter Modelica.SIunits.MassFlowRate Qmin=1.e-6     "Mass flow when the valve is closed";
  parameter Boolean continuous_flow_reversal=false     "true: continuous flow reversal - false: discontinuous flow reversal";
  parameter Integer fluid=1 "1: water/steam - 2: C3H3F5";
  parameter Modelica.SIunits.Density p_rho=0 "If > 0, fixed fluid density";
  parameter Integer mode=0     "IF97 region. 1:liquid - 2:steam - 4:saturation line - 0:automatic";

  WaterSteam.Connectors.FluidInletI fluidInletI;
  WaterSteam.Connectors.FluidOutletI fluidOutletI;
  WaterSteam.PressureLosses.SwitchValve switchValve "standard switch valve";
  Modelica_StateGraph2.Step Open(
    nOut=2,
    nIn=3,
    initialStep=true) "open valve";
  Transitions_detailed.newTransitions.MultiTransition Inadv_Opening(immediate=false,
      kind=2) "inadvertent opening of the valve";
  Modelica_StateGraph2.Step StuckClosed(nIn=2, nOut=1) "stuck closed valve";
```

```
Modelica_StateGraph2.Step StuckOpen(nIn=2, nOut=1) "stuck open valve";
Modelica_StateGraph2.Step Closed(nIn=1, nOut=2) "closed valve";
Transitions_detailed.newTransitions.TransitionOnRequest Req_Closing     "requested closing of the valve";
Transitions_detailed.newTransitions.TransitionOnRequest Req_Opening     "requested opening of the valve";
Transitions_detailed.newTransitions.MultiTransition Inadv_Closing(immediate=false,
    kind=2) "inadvertent closing of the valve";
Transitions_detailed.newTransitions.MultiTransition RepFromClosed(immediate=false,
    kind=3) "reparation from stuck closed state to open state";
Transitions_detailed.newTransitions.MultiTransition RepFromOpen(immediate=false,
    kind=3) "reparation from stuck open state to open state";
InstrumentationAndControl.Connectors.InputLogical requestOpening     "requested opening of the valve : true -> open";
WaterSteam.PressureLosses.ThreeWayValve     splitter2_1     "control of the leaking (1 = no leak)";
InstrumentationAndControl.Blocks.NonLineaire.Selecteur              selecteur;
InstrumentationAndControl.Blocks.Sources.Constante              constante(k=1);
InstrumentationAndControl.Blocks.Sources.Constante              constante1(k=0.8);
WaterSteam.BoundaryConditions.SinkP             sink1 "sink for the leak";
WaterSteam.PressureLosses.SingularPressureLoss
  singularPressureLoss1;
Modelica_StateGraph2.Step Leak_pre(nIn=1, nOut=1) "leak before valve";
Transitions_detailed.newTransitions.MultiTransition pre_leaking(immediate=
     false, kind=2) "apparition of a leak before the valve";
Transitions_detailed.newTransitions.MultiTransition RepFromPreLeak(immediate=
     false, kind=3) "reparation from pre-leaking state to normal state";
Modelica_StateGraph2.Step OK_preLeak(
  nOut=1,
  nIn=1,
  initialStep=true) "no pre-leak";
WaterSteam.Volumes.VolumeA volumeA;
WaterSteam.Volumes.VolumeC volumeC;
WaterSteam.PressureLosses.ControlValve controlValve;
InstrumentationAndControl.Blocks.NonLineaire.Selecteur              selecteur1;
InstrumentationAndControl.Blocks.Sources.Constante              constante2(k=0);
InstrumentationAndControl.Blocks.Sources.Constante              constante3(k=0.2);
Modelica_StateGraph2.Step Leak_intern(nIn=1, nOut=1)     "leak inside the valve when closed";
Transitions_detailed.newTransitions.MultiTransition leaking1(
                                                    immediate=false,
    kind=2) "inadvertent opening of the valve";
Transitions_detailed.newTransitions.MultiTransition RepFromInternLeak(immediate=
     false, kind=3) "reparation from intern-leaking state to normal state";
Modelica_StateGraph2.Step OK_internLeak(
  nOut=1,
  nIn=1,
  initialStep=true) "no intern-leak";
Modelica_StateGraph2.Step Leak_post(nIn=1, nOut=1) "leak after the valve";
Transitions_detailed.newTransitions.MultiTransition post_leak(immediate=
     false, kind=2) "apparition of a leak after the valve";
Transitions_detailed.newTransitions.MultiTransition RepFromPostLeak(immediate=
```

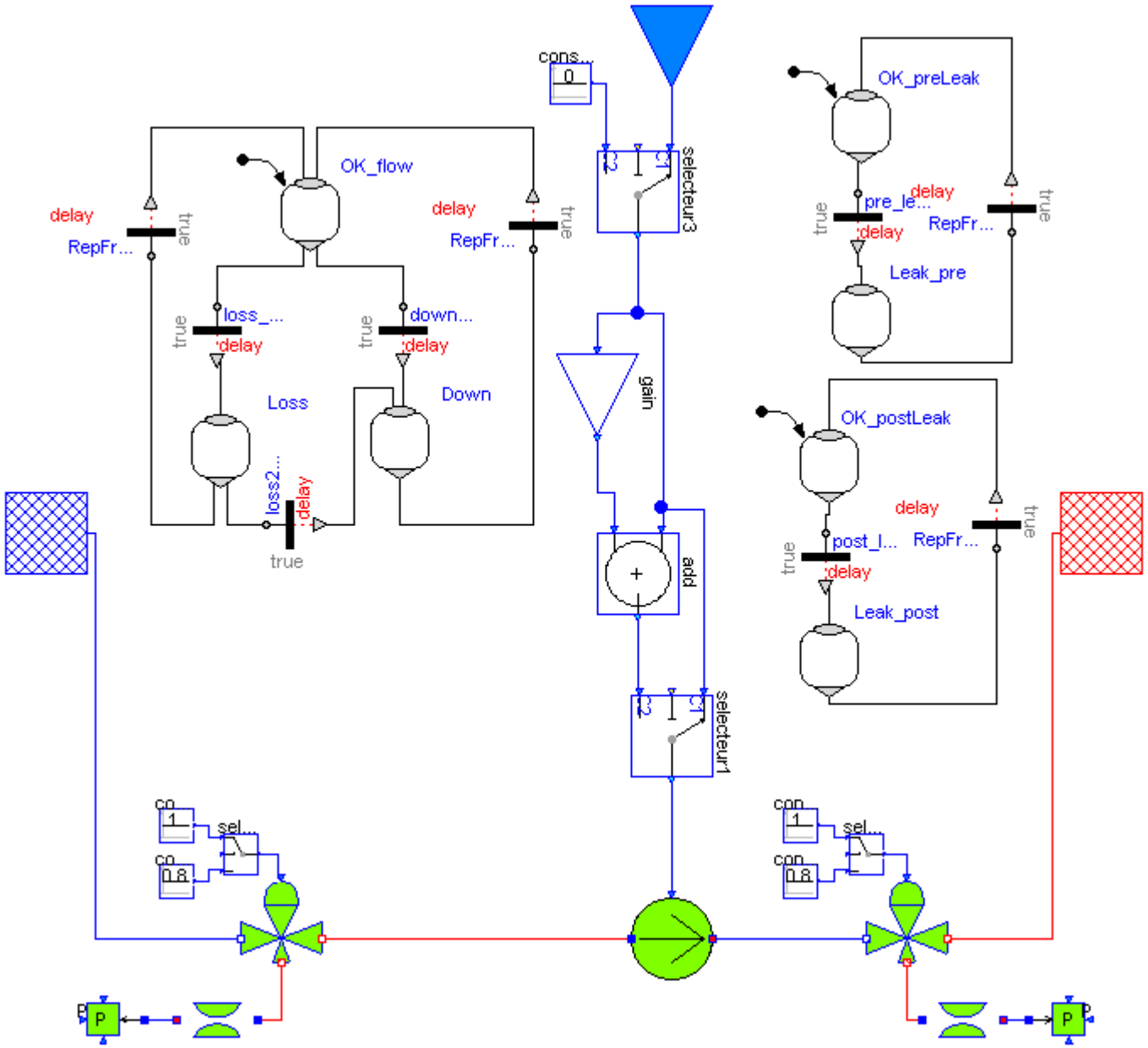
```
      false, kind=3) "reparation from post-leaking state to normal state";
  Modelica_StateGraph2.Step OK_postLeak(
    nOut=1,
    nIn=1,
    initialStep=true) "no post-leak";
  WaterSteam.PressureLosses.ThreeWayValve     splitter2_2     "control of the leaking (1 = no leak)";
  WaterSteam.BoundaryConditions.SinkP             sink2 "sink for the leak";
  WaterSteam.PressureLosses.SingularPressureLoss
    singularPressureLoss2;
  InstrumentationAndControl.Blocks.NonLineaire.Selecteur            selecteur2;
  InstrumentationAndControl.Blocks.Sources.Constante             constante4(
                                                                          k=1);
  InstrumentationAndControl.Blocks.Sources.Constante             constante5(k=0.8);
equation
  switchValve.Ouv.signal = Open.active or StuckOpen.active;
  Req_Opening.request = requestOpening.signal;
  Req_Closing.request = not requestOpening.signal;
  // leaks part
  selecteur.uCond.signal = not Leak_pre.active;
  selecteur1.uCond.signal = not Leak_intern.active;
  selecteur2.uCond.signal = not Leak_post.active;
  connect(Open.outPort[1], Req_Closing.inPort);
  connect(Req_Closing.goodPath, Closed.inPort[1]);
  connect(Req_Closing.failingPath, StuckOpen.inPort[1]);
  connect(Closed.outPort[2], Req_Opening.inPort);
  connect(Req_Opening.goodPath, Open.inPort[3]);

  connect(Req_Opening.failingPath, StuckClosed.inPort[1]);
  connect(Inadv_Closing.inPort, Open.outPort[2]);
  connect(Inadv_Closing.outPort, StuckClosed.inPort[2]);
  connect(StuckClosed.outPort[1], RepFromClosed.inPort);
  connect(RepFromClosed.outPort, Open.inPort[2]);
  connect(Inadv_Opening.inPort, Closed.outPort[1]);
  connect(Inadv_Opening.outPort, StuckOpen.inPort[2]);
  connect(StuckOpen.outPort[1], RepFromOpen.inPort);
  connect(RepFromOpen.outPort, Open.inPort[1]);
  connect(constante1.y,selecteur. u2);
  connect(constante.y,selecteur. u1);
  connect(singularPressureLoss1.C2,sink1. C);
  connect(pre_leaking.outPort, Leak_pre.inPort[1]);
  connect(Leak_pre.outPort[1], RepFromPreLeak.inPort);
  connect(pre_leaking.inPort, OK_preLeak.outPort[1]);
  connect(RepFromPreLeak.outPort, OK_preLeak.inPort[1]);
  connect(selecteur.y, splitter2_1.Ouv);
  connect(splitter2_1.C1, fluidInletI);
  connect(splitter2_1.C3, singularPressureLoss1.C1);
  connect(splitter2_1.C2, volumeA.Ce1);
```

```
  connect(volumeA.Cs1, switchValve.C1);
  connect(switchValve.C2, volumeC.Ce2);

  connect(controlValve.C1, volumeA.Cs2);
  connect(volumeC.Ce3, controlValve.C2);
  connect(constante3.y, selecteur1.u2);
  connect(constante2.y, selecteur1.u1);
  connect(selecteur1.y, controlValve.Ouv);
  connect(Leak_intern.outPort[1], RepFromInternLeak.inPort);
  connect(RepFromInternLeak.outPort, OK_internLeak.inPort[1]);
  connect(OK_internLeak.outPort[1], leaking1.inPort);
  connect(leaking1.outPort, Leak_intern.inPort[1]);
  connect(Leak_post.outPort[1], RepFromPostLeak.inPort);
  connect(RepFromPostLeak.outPort, OK_postLeak.inPort[1]);
  connect(OK_postLeak.outPort[1], post_leak.inPort);
  connect(post_leak.outPort, Leak_post.inPort[1]);
  connect(volumeC.Cs, splitter2_2.C1);
  connect(splitter2_2.C2, fluidOutletI);
  connect(singularPressureLoss2.C2,sink2. C);
  connect(splitter2_2.C3, singularPressureLoss2.C1);
  connect(constante5.y, selecteur2.u2);
  connect(constante4.y, selecteur2.u1);
  connect(selecteur2.y, splitter2_2.Ouv);
end SwitchValve8;
```



First_Thermo.ThermoSysPro.Experiments.Pump2

StaticCentrifugalPump with failures


cons...
selecteur3
OK_flow
delay
RepFr...
true
loss_...
down...
Loss
Down
loss2...
gain
add
OK_preLeak
pre_le...
Leak_pre
OK_postLeak
post_l...
Leak_post
selecteur1
sel...
P

Information

This pump can lose all power, lose 10% of its power, have a leak on the blue side, have a leak on the red side.

Parameters

Type	Name	Default	Description
AngularVelocity_rpm	VRot	1400	Fixed rotational speed (active if fixed_rot_or_power=1 and rpm_or_mpower connector not connected) [rev/min]
Power	MPower	0.1e6	Fixed mechanical power (active if fixed_rot_or_power=2 and rpm_or_mpower connector not connected) [W]
AngularVelocity_rpm	VRotn	1400	Nominal rotational speed [rev/min]
Real	rm	0.85	Product of the pump mechanical and electrical efficiencies
Integer	fixed_rot_or_power	1	1: fixed rotational speed - 2: fixed mechanical power
Boolean	adiabatic_compression	false	true: compression at constant enthalpy - false: compression with varying enthalpy
Boolean	continuous_flow_reversal	false	true: continuous flow reversal - false: discontinuous flow reversal
Integer	fluid	1	1: water/steam - 2: C3H3F5
Density	p_rho	0	If > 0, fixed fluid density [kg/m3]
Integer	mode	0	IF97 region. 1:liquid - 2:steam - 4:saturation line - 0:automatic
Real	a1	-88.67	x^2 coef. of the pump characteristics hn = f(vol_flow) (s2/m5)
Real	a2	0	x coef. of the pump characteristics hn = f(vol_flow) (s/m2)
Real	a3	43.15	Constant coef. of the pump characteristics hn = f(vol_flow) (m)
Real	b1	-3.7751	x^2 coef. of the pump efficiency characteristics rh = f(vol_flow) (s2/m6)
Real	b2	3.61	x coef. of the pump efficiency characteristics rh = f(vol_flow) (s/m3)
Real	b3	-0.0075464	Constant coef. of the pump efficiency characteristics rh = f(vol_flow) (s.u.)

Connectors

Type	Name	Description
FluidInletI	fluidInletI	
FluidOutletI	fluidOutletI	
InputReal	rpm_or_mpower	

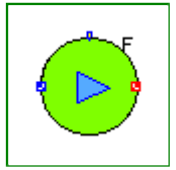
Modelica definition

```
model Pump2 "StaticCentrifugalPump with failures"
  parameter ThermoSysPro.Units.AngularVelocity_rpm VRot=1400     "Fixed rotational speed (active if fixed_rot_or_power=1 and rpm_or_mpo
  parameter Modelica.SIunits.Power MPower=0.1e6     "Fixed mechanical power (active if fixed_rot_or_power=2 and rpm_or_mpower connector
  parameter ThermoSysPro.Units.AngularVelocity_rpm VRotn=1400     "Nominal rotational speed";
  parameter Real rm=0.85     "Product of the pump mechanical and electrical efficiencies";
  parameter Integer fixed_rot_or_power=1     "1: fixed rotational speed - 2: fixed mechanical power";
  parameter Boolean adiabatic_compression=false     "true: compression at constant enthalpy - false: compression with varying enthalpy"
  parameter Boolean continuous_flow_reversal=false     "true: continuous flow reversal - false: discontinuous flow reversal";
  parameter Integer fluid=1 "1: water/steam - 2: C3H3F5";
  parameter Modelica.SIunits.Density p_rho=0 "If > 0, fixed fluid density";
  parameter Integer mode=0     "IF97 region. 1:liquid - 2:steam - 4:saturation line - 0:automatic";

  parameter Real a1=-88.67     "x^2 coef. of the pump characteristics hn = f(vol_flow) (s2/m5)";
  parameter Real a2=0     "x coef. of the pump characteristics hn = f(vol_flow) (s/m2)";
  parameter Real a3=43.15     "Constant coef. of the pump characteristics hn = f(vol_flow) (m)";

  parameter Real b1=-3.7751     "x^2 coef. of the pump efficiency characteristics rh = f(vol_flow) (s2/m6)";
  parameter Real b2=3.61     "x coef. of the pump efficiency characteristics rh = f(vol_flow) (s/m3)";
  parameter Real b3=-0.0075464     "Constant coef. of the pump efficiency characteristics rh = f(vol_flow) (s.u.)";

  WaterSteam.Connectors.FluidInletI fluidInletI;
  WaterSteam.Connectors.FluidOutletI fluidOutletI;
  WaterSteam.Machines.StaticCentrifugalPump pump(
    VRot=VRot,
    MPower=MPower,
    VRotn=VRotn,
    rm=rm,
    fixed_rot_or_power=fixed_rot_or_power,
    adiabatic_compression=adiabatic_compression,
    continuous_flow_reversal=continuous_flow_reversal,
    fluid=fluid,
    p_rho=p_rho,
    mode=mode,
    a1=a1,
    a2=a2,
    a3=a3,
    b1=b1,
    b2=b2,
    b3=b3) "standard static centrifugal pump";
  WaterSteam.PressureLosses.ThreeWayValve     splitter2_1     "control of the leaking (1 = no leak)";
  InstrumentationAndControl.Blocks.NonLineaire.Selecteur              selecteur;
  InstrumentationAndControl.Blocks.Sources.Constante              constante(k=1);
  InstrumentationAndControl.Blocks.Sources.Constante              constante1(k=0.8);
  WaterSteam.BoundaryConditions.SinkP             sink1 "sink for the leak";
  WaterSteam.PressureLosses.SingularPressureLoss
```

```
  singularPressureLoss1;
Modelica_StateGraph2.Step Leak_pre(nIn=1, nOut=1) "leak before valve";
Transitions_detailed.newTransitions.MultiTransition pre_leaking(immediate=
      false, kind=2) "apparition of a leak before the valve";
Transitions_detailed.newTransitions.MultiTransition RepFromPreLeak(immediate=
     false, kind=3) "reparation from pre-leaking state to normal state";
Modelica_StateGraph2.Step OK_preLeak(
  nOut=1,
  nIn=1,
  initialStep=true) "no pre-leak";
Modelica_StateGraph2.Step Leak_post(nIn=1, nOut=1) "leak after the valve";
Transitions_detailed.newTransitions.MultiTransition post_leak(immediate=
      false, kind=2) "apparition of a leak after the valve";
Transitions_detailed.newTransitions.MultiTransition RepFromPostLeak(immediate=
     false, kind=3) "reparation from post-leaking state to normal state";
Modelica_StateGraph2.Step OK_postLeak(
  nOut=1,
  nIn=1,
  initialStep=true) "no post-leak";
WaterSteam.PressureLosses.ThreeWayValve     splitter2_2     "control of the leaking (1 = no leak)";
WaterSteam.BoundaryConditions.SinkP             sink2 "sink for the leak";
WaterSteam.PressureLosses.SingularPressureLoss
  singularPressureLoss2;
InstrumentationAndControl.Blocks.NonLineaire.Selecteur              selecteur2;
InstrumentationAndControl.Blocks.Sources.Constante              constante4(
                                                                        k=1);
InstrumentationAndControl.Blocks.Sources.Constante              constante5(k=0.8);
InstrumentationAndControl.Connectors.InputReal rpm_or_mpower;
InstrumentationAndControl.Blocks.Math.Add add(k2=-1);
InstrumentationAndControl.Blocks.Math.Gain gain(Gain=0.1);
InstrumentationAndControl.Blocks.NonLineaire.Selecteur selecteur1;
Modelica_StateGraph2.Step Loss(nIn=1, nOut=2) "loss";
Transitions_detailed.newTransitions.MultiTransition loss_pump(immediate=false,
    kind=2) "apparition of a loss";
Transitions_detailed.newTransitions.MultiTransition RepFromLoss(immediate=false,
    kind=3) "reparation from loss state to normal state";
Modelica_StateGraph2.Step OK_flow(
  nIn=2,
  initialStep=true,
  nOut=2) "no loss";
Modelica_StateGraph2.Step Down(nIn=2, nOut=1) "down";
Transitions_detailed.newTransitions.MultiTransition down_pump1(
                                                        immediate=false,
    kind=2) "failing of the pump";
Transitions_detailed.newTransitions.MultiTransition RepFromDown(immediate=false,
    kind=3) "reparation from down state to normal state";
InstrumentationAndControl.Blocks.NonLineaire.Selecteur selecteur3;
```

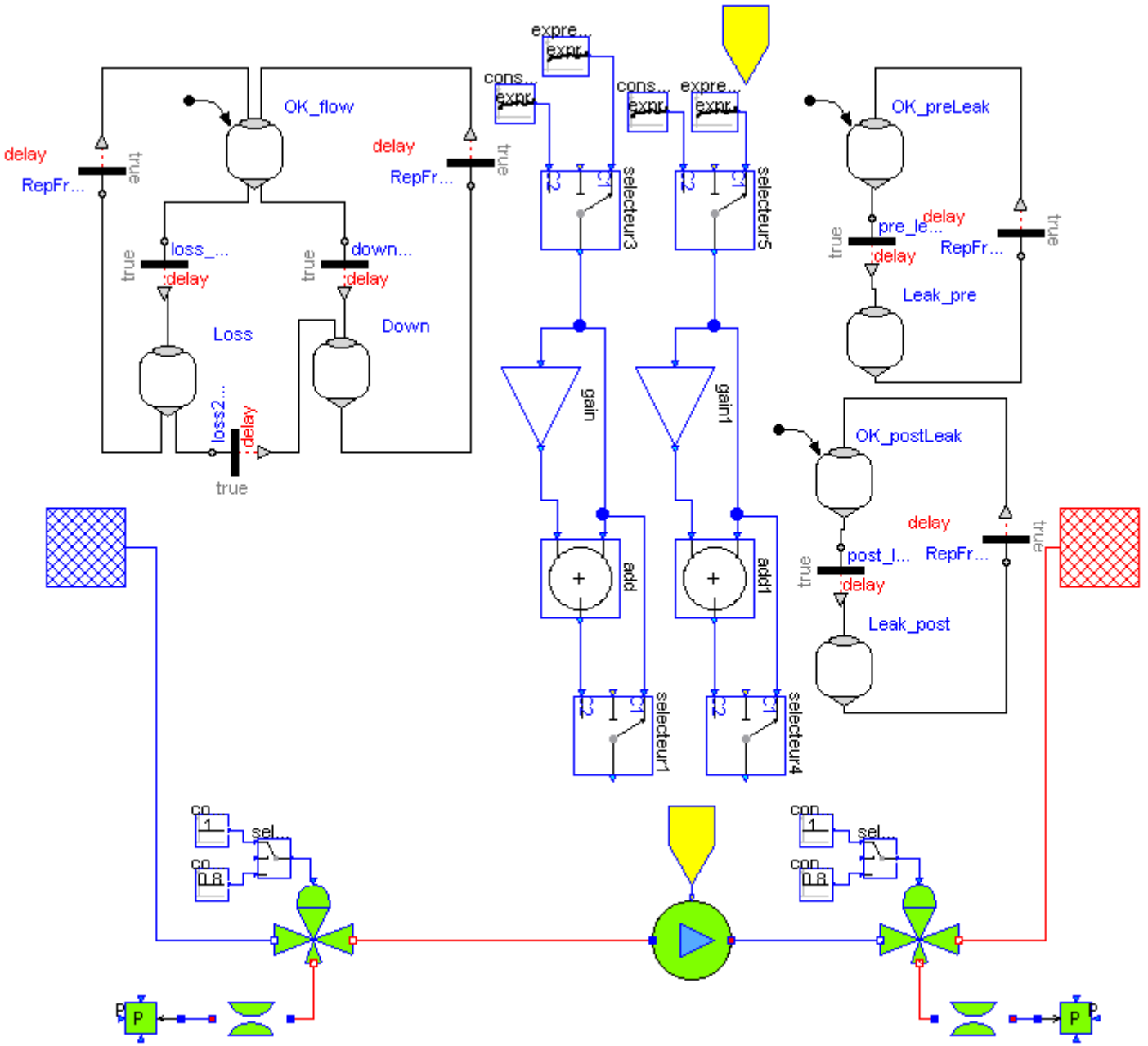
```
  InstrumentationAndControl.Blocks.Sources.Constante              constante2(k=0);
  Transitions_detailed.newTransitions.MultiTransition loss2down_pump(immediate=false,
      kind=2) "apparition of a loss";
equation
  // failures part
  selecteur.uCond.signal = not Leak_pre.active;
  selecteur2.uCond.signal = not Leak_post.active;
  selecteur1.uCond.signal = not Loss.active;
  selecteur3.uCond.signal = not Down.active;

  // part of the classic model for the pump
  if (cardinality(rpm_or_mpower) == 0) then
    if (fixed_rot_or_power == 1) then
      rpm_or_mpower.signal = VRot;
    elseif (fixed_rot_or_power == 2) then
      rpm_or_mpower.signal = MPower;
    else
      assert(false, "StaticCentrifugalPump: incorrect option");
    end if;
  end if;

  connect(constante1.y,selecteur. u2);
  connect(constante.y,selecteur. u1);
  connect(singularPressureLoss1.C2,sink1. C);
  connect(pre_leaking.outPort, Leak_pre.inPort[1]);
  connect(Leak_pre.outPort[1], RepFromPreLeak.inPort);
  connect(pre_leaking.inPort, OK_preLeak.outPort[1]);
  connect(RepFromPreLeak.outPort, OK_preLeak.inPort[1]);
  connect(selecteur.y, splitter2_1.Ouv);
  connect(splitter2_1.C1, fluidInletI);
  connect(splitter2_1.C3, singularPressureLoss1.C1);

  connect(Leak_post.outPort[1], RepFromPostLeak.inPort);
  connect(RepFromPostLeak.outPort, OK_postLeak.inPort[1]);
  connect(OK_postLeak.outPort[1], post_leak.inPort);
  connect(post_leak.outPort, Leak_post.inPort[1]);
  connect(splitter2_2.C2, fluidOutletI);
  connect(singularPressureLoss2.C2,sink2. C);
  connect(splitter2_2.C3, singularPressureLoss2.C1);
  connect(constante5.y, selecteur2.u2);
  connect(constante4.y, selecteur2.u1);
  connect(selecteur2.y, splitter2_2.Ouv);
  connect(splitter2_1.C2, pump.C1);
  connect(pump.C2, splitter2_2.C1);
  connect(gain.y, add.u2);
  connect(add.y, selecteur1.u2);
  connect(selecteur1.y, pump.rpm_or_mpower);
```

```
  connect(loss_pump.outPort, Loss.inPort[1]);
  connect(Loss.outPort[1], RepFromLoss.inPort);
  connect(RepFromLoss.outPort, OK_flow.inPort[1]);
  connect(loss_pump.inPort, OK_flow.outPort[1]);
  connect(Down.inPort[2], down_pump1.outPort);
  connect(down_pump1.inPort, OK_flow.outPort[2]);
  connect(Down.outPort[1], RepFromDown.inPort);
  connect(RepFromDown.outPort, OK_flow.inPort[2]);
  connect(selecteur3.u1, rpm_or_mpower);
  connect(selecteur3.y, gain.u);
  connect(add.u1, gain.u);
  connect(selecteur1.u1, gain.u);
  connect(selecteur3.u2, constante2.y);
  connect(loss2down_pump.inPort, Loss.outPort[2]);
  connect(loss2down_pump.outPort, Down.inPort[1]);
end Pump2;
```



First_Thermo.ThermoSysPro.Experiments.**PumpN15**

CentrifugalPump with failures


OK_flow
delay
RepFr...
true
loss_...
delay
down...
Loss
Down
loss2...
expre...
cons...
selecteur3
selecteur5
gain
gain1
add
add1
selecteur1
selecteur4
OK_preLeak
pre_le...
Leak_pre
OK_postLeak
post_l...
Leak_post
sel...
P

Information

This pump can lose all power, lose 10% of its power, have a leak on the blue side, have a leak on the red side.

Parameters

Type	Name	Default	Description
AngularVelocity_rpm	N	1400	Pump angular velocity in rpm (active if input M is not connected) [rev/min]
AngularVelocity_rpm	N_nom	1400	Nominal angular velocity in rpm [rev/min]
MomentOfInertia	J	10	Rotating masses moment of inertia (active if dynamic_mech_equation=true) [kg.m2]
Volume	V	1	Pump volume (active if dynamic_energy_balance=true) [m3]
Boolean	dynamic_mech_equation	false	true: dynamic mechanical equation - false: static mechanical equation (active if input M is connected)
Boolean	dynamic_energy_balance	false	true: dynamic energy balance equation - false: static energy balance equation
Boolean	continuous_flow_reversal	false	true: continuous flow reversal - false: discontinuous flow reversal
Integer	fluid	1	1: water/steam - 2: C3H3F5
Density	p_rho	0	If > 0, fixed fluid density [kg/m3]
Integer	mode	0	IF97 region. 1:liquid - 2:steam - 4:saturation line - 0:automatic
Integer	mode_car	2	1:nominal values and coef. c given by parameters - 2:nominal values and coef. c computed from semi-parabolic characteristics
Integer	mode_car_hn	2	1:complete pump head characteristics - 2:semi-parabolic pump head characteristics
Integer	mode_car_Cr	2	1:complete torque characteristics - 2:analytic formula
VolumeFlowRate	Qv_nom_p	0.4781	Nominal volumetric flow (active if mode_car=1) [m3/s]
Height	hn_nom_p	22.879	Nominal pump head (active if mode_car=1) [m]
Height	rh_nom_p	0.863	Nominal pump efficiency (active if mode_car=1) [m]
Real	F_t[:]	{0.634,0.643,0.646,0.640,0.6...	Head characteristics (active if mode_car_hn=1)
Real	G_t[:]	{-0.684,-0.547,-0.414,-0.292...	Torque characteristics (active if mode_car_Cr=1)
Real	c_p	1.288	Dimensionless coef. of the semi-parabolic pump head characteristics (active if mode_car=1 and mode_car_hn=2)
Real	hn_coef[2]	{-88.67,43.15}	Coef. of the semi-parabolic pump head characteristics (active if mode_car=2)
Real	rh_coef[2]	{-3.7751,3.61}	Coef. of the parabolic pump efficiency characteristics (active if mode_car=2)

Connectors

Type	Name	Description
FluidInletI	fluidInletI	
FluidOutletI	fluidOutletI	
MechanichalTorque	M	
MechanichalTorque	M1	

Modelica definition

```
model PumpN15 "CentrifugalPump with failures"
  parameter ThermoSysPro.Units.AngularVelocity_rpm N=1400     "Pump angular velocity in rpm (active if input M is not connected)";
  parameter ThermoSysPro.Units.AngularVelocity_rpm N_nom=1400     "Nominal angular velocity in rpm";
  parameter Modelica.SIunits.MomentOfInertia J=10     "Rotating masses moment of inertia (active if dynamic_mech_equation=true)";
  parameter Modelica.SIunits.Volume V=1     "Pump volume (active if dynamic_energy_balance=true)";
  parameter Boolean dynamic_mech_equation=false     "true: dynamic mechanical equation - false: static mechanical equation (active if i
  parameter Boolean dynamic_energy_balance=false     "true: dynamic energy balance equation - false: static energy balance equation";
  parameter Boolean continuous_flow_reversal=false     "true: continuous flow reversal - false: discontinuous flow reversal";
  parameter Integer fluid=1 "1: water/steam - 2: C3H3F5";
  parameter Modelica.SIunits.Density p_rho=0 "If > 0, fixed fluid density";
  parameter Integer mode=0     "IF97 region. 1:liquid - 2:steam - 4:saturation line - 0:automatic";

  parameter Integer mode_car=2     "1:nominal values and coef. c given by parameters - 2:nominal values and coef. c computed from semi-
  parameter Integer mode_car_hn=2     "1:complete pump head characteristics - 2:semi-parabolic pump head characteristics";
  parameter Integer mode_car_Cr=2     "1:complete torque characteristics - 2:analytic formula";

  parameter Modelica.SIunits.VolumeFlowRate Qv_nom_p=0.4781     "Nominal volumetric flow (active if mode_car=1)";
  parameter Modelica.SIunits.Height hn_nom_p=22.879     "Nominal pump head (active if mode_car=1)";
  parameter Modelica.SIunits.Height rh_nom_p=0.863     "Nominal pump efficiency (active if mode_car=1)";

  parameter Real F_t[:]={ 0.634, 0.643, 0.646, 0.640, 0.629, 0.613, 0.595, 0.575, 0.552, 0.533, 0.516, 0.505,
                          0.504, 0.510, 0.512, 0.522, 0.539, 0.559, 0.580, 0.601, 0.630, 0.662, 0.692, 0.722,
                          0.753, 0.782, 0.808, 0.832, 0.857, 0.879, 0.904, 0.930, 0.959, 0.996, 1.027, 1.060,
                          1.090, 1.124, 1.165, 1.204, 1.238, 1.258, 1.271, 1.282, 1.288, 1.281, 1.260, 1.225,
                          1.172, 1.107, 1.031, 0.942, 0.842, 0.733, 0.617, 0.500, 0.368, 0.240, 0.125, 0.011,
                         -0.102,-0.168,-0.255,-0.342,-0.423,-0.494,-0.556,-0.620,-0.655,-0.670,-0.670,-0.660,
                         -0.655,-0.640,-0.600,-0.570,-0.520,-0.470,-0.430,-0.360,-0.275,-0.160,-0.040, 0.130,
                          0.295, 0.430, 0.550, 0.620, 0.634}     "Head characteristics (active if mode_car_hn=1)";

  parameter Real G_t[:]={-0.684,-0.547,-0.414,-0.292,-0.187,-0.105,-0.053,-0.012, 0.042, 0.097, 0.156, 0.227,
                          0.300, 0.371, 0.444, 0.522, 0.596, 0.672, 0.738, 0.763, 0.797, 0.837, 0.865, 0.883,
                          0.886, 0.877, 0.859, 0.838, 0.804, 0.758, 0.703, 0.645, 0.583, 0.520, 0.454, 0.408,
                          0.370, 0.343, 0.331, 0.329, 0.338, 0.354, 0.372, 0.405, 0.450, 0.486, 0.520, 0.552,
                          0.579, 0.603, 0.616, 0.617, 0.606, 0.582, 0.546, 0.500, 0.432, 0.360, 0.288, 0.214,
```

```
                        0.123, 0.037,-0.053,-0.161,-0.248,-0.314,-0.372,-0.580,-0.740,-0.880,-1.000,-1.120,
                       -1.250,-1.370,-1.490,-1.590,-1.660,-1.690,-1.770,-1.650,-1.590,-1.520,-1.420,-1.320,
                       -1.230,-1.100,-0.980,-0.820, -0.684}     "Torque characteristics (active if mode_car_Cr=1)";

  parameter Real c_p=1.288     "Dimensionless coef. of the semi-parabolic pump head characteristics (active if mode_car=1 and mode_car_
  constant Real b=2     "Dimensionless coef. of the parabolic pump efficiency characteristics";

  parameter Real hn_coef[2]={-88.67, 43.15}     "Coef. of the semi-parabolic pump head characteristics (active if mode_car=2)";
  parameter Real rh_coef[2]={-3.7751, 3.61}     "Coef. of the parabolic pump efficiency characteristics (active if mode_car=2)";

protected
  constant Modelica.SIunits.Acceleration g=Modelica.Constants.g_n     "Gravity constant";
  constant Real pi=Modelica.Constants.pi "pi";
  parameter Modelica.SIunits.AngularVelocity w_a_min=1.e-4     "Small angular velocity";
  parameter Modelica.SIunits.VolumeFlowRate Qv_a_min=1.e-4     "Small volume flow rate";
  parameter Real rh_min=0.05 "Minimum efficiency";
  parameter Modelica.SIunits.MassFlowRate Qeps=1.e-3     "Small mass flow for continuous flow reversal";
  parameter Boolean dyn_mech_equation=((cardinality(M) <> 0) and dynamic_mech_equation);

public
  WaterSteam.PressureLosses.ThreeWayValve     splitter2_1     "control of the leaking (1 = no leak)";
  InstrumentationAndControl.Blocks.NonLineaire.Selecteur              selecteur;
  InstrumentationAndControl.Blocks.Sources.Constante              constante(k=1);
  InstrumentationAndControl.Blocks.Sources.Constante              constante1(k=0.8);
  WaterSteam.BoundaryConditions.SinkP              sink1 "sink for the leak";
  WaterSteam.PressureLosses.SingularPressureLoss
    singularPressureLoss1;
  Modelica_StateGraph2.Step Leak_pre(nIn=1, nOut=1) "leak before valve";
  Transitions_detailed.newTransitions.MultiTransition pre_leaking(immediate=
        false, kind=2) "apparition of a leak before the valve";
  Transitions_detailed.newTransitions.MultiTransition RepFromPreLeak(immediate=
       false, kind=3) "reparation from pre-leaking state to normal state";
  Modelica_StateGraph2.Step OK_preLeak(
    nOut=1,
    nIn=1,
    initialStep=true) "no pre-leak";
  Modelica_StateGraph2.Step Leak_post(nIn=1, nOut=1) "leak after the valve";
  Transitions_detailed.newTransitions.MultiTransition post_leak(immediate=
        false, kind=2) "apparition of a leak after the valve";
  Transitions_detailed.newTransitions.MultiTransition RepFromPostLeak(immediate=
       false, kind=3) "reparation from post-leaking state to normal state";
  Modelica_StateGraph2.Step OK_postLeak(
    nOut=1,
    nIn=1,
    initialStep=true) "no post-leak";
  WaterSteam.PressureLosses.ThreeWayValve     splitter2_2     "control of the leaking (1 = no leak)";
  WaterSteam.BoundaryConditions.SinkP              sink2 "sink for the leak";
```

```
WaterSteam.PressureLosses.SingularPressureLoss
  singularPressureLoss2;
InstrumentationAndControl.Blocks.NonLineaire.Selecteur             selecteur2;
InstrumentationAndControl.Blocks.Sources.Constante             constante4(
                                                                       k=1);
InstrumentationAndControl.Blocks.Sources.Constante             constante5(k=0.8);
InstrumentationAndControl.Blocks.Math.Add add(k2=-1);
InstrumentationAndControl.Blocks.Math.Gain gain(Gain=0.1);
InstrumentationAndControl.Blocks.NonLineaire.Selecteur selecteur1;
Modelica_StateGraph2.Step Loss(nIn=1, nOut=2) "loss";
Transitions_detailed.newTransitions.MultiTransition loss_pump(immediate=false,
    kind=2) "apparition of a loss";
Transitions_detailed.newTransitions.MultiTransition RepFromLoss(immediate=false,
    kind=3) "reparation from loss state to normal state";
Modelica_StateGraph2.Step OK_flow(
  nIn=2,
  initialStep=true,
  nOut=2) "no loss";
Modelica_StateGraph2.Step Down(nIn=2, nOut=1) "down";
Transitions_detailed.newTransitions.MultiTransition down_pump1(
                                                          immediate=false,
    kind=2) "failing of the pump";
Transitions_detailed.newTransitions.MultiTransition RepFromDown(immediate=false,
    kind=3) "reparation from down state to normal state";
InstrumentationAndControl.Blocks.NonLineaire.Selecteur selecteur3;
Expression                                                     constante2(exp=0.01*
      M.Ctr);
Transitions_detailed.newTransitions.MultiTransition loss2down_pump(immediate=false,
    kind=2) "apparition of a loss";
WaterSteam.Machines.CentrifugalPump centrifugalPump(
  N=N,
  N_nom=N_nom,
  J=J,
  V=V,
  dynamic_mech_equation=dynamic_mech_equation,
  dynamic_energy_balance=dynamic_energy_balance,
  continuous_flow_reversal=continuous_flow_reversal,
  fluid=fluid,
  p_rho=p_rho,
  mode=mode,
  mode_car=mode_car,
  mode_car_hn=mode_car_hn,
  mode_car_Cr=mode_car_Cr,
  Qv_nom_p=Qv_nom_p,
  hn_nom_p=hn_nom_p,
  rh_nom_p=rh_nom_p,
  F_t=F_t,
```

```
    G_t=G_t,
    c_p=c_p,
    hn_coef=hn_coef,
    rh_coef=rh_coef);
  InstrumentationAndControl.Blocks.Math.Add add1(
                                          k2=-1);
  InstrumentationAndControl.Blocks.Math.Gain gain1(
                                           Gain=0.1);
  InstrumentationAndControl.Blocks.NonLineaire.Selecteur selecteur4;
  InstrumentationAndControl.Blocks.NonLineaire.Selecteur selecteur5;
public
  WaterSteam.Connectors.FluidInletI fluidInletI;
  WaterSteam.Connectors.FluidOutletI fluidOutletI;
  ElectroMechanics.Connectors.MechanichalTorque M;
  ElectroMechanics.Connectors.MechanichalTorque M1;
  Expression expression(exp=M.Ctr);
  Expression expression1(exp=M.w);
  Expression                                              constante3(exp=0.01*
      M.w);
equation
  //failures part
  selecteur.uCond.signal = not Leak_pre.active;
  selecteur2.uCond.signal = not Leak_post.active;
  selecteur1.uCond.signal = not Loss.active;
  selecteur3.uCond.signal = not Down.active;
  selecteur4.uCond.signal = not Loss.active;
  selecteur5.uCond.signal = not Down.active;

  // interfacing
  M1.Ctr = selecteur1.y.signal;
  M1.w = selecteur4.y.signal;

  // part of the original model for the pump
  if (cardinality(M) == 0) then
    M.w = pi/30*N;
  end if;

  connect(constante1.y,selecteur. u2);
  connect(constante.y,selecteur. u1);
  connect(singularPressureLoss1.C2,sink1. C);
  connect(pre_leaking.outPort, Leak_pre.inPort[1]);
  connect(Leak_pre.outPort[1], RepFromPreLeak.inPort);
  connect(pre_leaking.inPort, OK_preLeak.outPort[1]);
  connect(RepFromPreLeak.outPort, OK_preLeak.inPort[1]);
  connect(selecteur.y, splitter2_1.Ouv);
  connect(splitter2_1.C3, singularPressureLoss1.C1);
```

```
  connect(Leak_post.outPort[1], RepFromPostLeak.inPort);
  connect(RepFromPostLeak.outPort, OK_postLeak.inPort[1]);
  connect(OK_postLeak.outPort[1], post_leak.inPort);
  connect(post_leak.outPort, Leak_post.inPort[1]);
  connect(singularPressureLoss2.C2,sink2. C);
  connect(splitter2_2.C3, singularPressureLoss2.C1);
  connect(constante5.y, selecteur2.u2);
  connect(constante4.y, selecteur2.u1);
  connect(selecteur2.y, splitter2_2.Ouv);
  connect(gain.y, add.u2);
  connect(add.y, selecteur1.u2);
  connect(loss_pump.outPort, Loss.inPort[1]);
  connect(Loss.outPort[1], RepFromLoss.inPort);
  connect(RepFromLoss.outPort, OK_flow.inPort[1]);
  connect(loss_pump.inPort, OK_flow.outPort[1]);
  connect(Down.inPort[2], down_pump1.outPort);
  connect(down_pump1.inPort, OK_flow.outPort[2]);
  connect(Down.outPort[1], RepFromDown.inPort);
  connect(RepFromDown.outPort, OK_flow.inPort[2]);
  connect(add.u1, gain.u);
  connect(selecteur1.u1, gain.u);
  connect(loss2down_pump.inPort, Loss.outPort[2]);
  connect(loss2down_pump.outPort, Down.inPort[1]);
  connect(splitter2_1.C2, centrifugalPump.C1);
  connect(centrifugalPump.C2, splitter2_2.C1);
  connect(selecteur3.y, gain.u);
  connect(constante2.y, selecteur3.u2);
  connect(gain1.y, add1.u2);
  connect(add1.y, selecteur4.u2);
  connect(add1.u1, gain1.u);
  connect(selecteur4.u1, gain1.u);
  connect(selecteur5.y, gain1.u);
  connect(fluidInletI, splitter2_1.C1);
  connect(splitter2_2.C2, fluidOutletI);
  connect(M1, centrifugalPump.M);
  connect(expression.y, selecteur3.u1);
  connect(expression1.y, selecteur5.u1);
  connect(constante3.y, selecteur5.u2);
end PumpN15;
```

Automatically generated Fri Sep 05 13:53:37 2014.