A Simple Test System Illustrating Load-Voltage Dynamics in Power Systems

Mats Larsson
Corporate Research
ABB Schweiz AG
5405 Baden-Dättwil
mailto:Mats.Larsson@ch.abb.com
Switzerland

March 27, 2003

1 Introduction

This test system is intended as a first step before tackling the official ABB Case study models. It is intended as an aid for non-power systems engineers to get acquainted with the dynamics of voltage collapse or to design alternative control schemes for the tap changer or load shedding. Because of it's simplicity, a detailed mathematical model can easily be derived.

2 Example System

A single load, single on-load tap changer (OLTC) system such as shown in Fig. 1 is considered. The example system represents a distribution bus fed through a tap changing transformer and a transmission system equivalent. To increase transfer limits, the distribution system has been compensated by a capacitor bank. Although simple, the system contains most of the principal components which affect voltage behaviour, especially after line fault or under heavy load conditions. Parameter data can be found in the Modelica code in Appendix A.

2.1 Open Loop System: Load and Network

Focusing on tap changer dynamics and therefore on a slower time scale (below ~ 0.1 Hz) than the one relevant to generator dynamics, the generator bus is modelled as an ideal voltage source. The load is modelled as an exponential

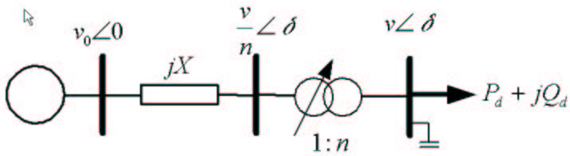


Figure 1: Example power system studied.

recovery load according to Karlsson and Hill (1994)

$$\dot{x}_p = -\frac{x_p}{T_p} + P_0(v^{\alpha_s} - v^{\alpha_t}) \quad (1)$$

$$P_d = (1-k)(x_p + P_0 v^{\alpha_t}) \quad (2)$$

$$\dot{x}_q = -\frac{x_q}{T_q} + Q_0(v^{\beta_s} - v^{\beta_t}) \quad (3)$$

$$Q_d = (1-k)(x_q + Q_0 v^{\beta_t}) \quad (4)$$

where v^{α_s} is the steady-state and v^{α_t} the transient voltage dependency. P_d is the actual active load power, P_0 the nominal[1] active power consumption and T_p is the active power recovery time constant. For the reactive load power, a similar model is used with corresponding characteristics v^{β_s}, v^{β_t} and time constant T_q. The transmission network is modelled as a pure reactance in series with an ideal transformer. The factor $(1-k)$ has been introduced as a scale factor on the load power to model load shedding.

The reactive production by the capacitor bank can be written

$$Q_{cap} = Bv^2 \quad (5)$$

Using basic theory, we can derive the following formulas for the active and reactive power supplied to the load by the transmission line and transformer[2] (Glover and Sarma, 1994)

$$P_d = -\frac{v_0 v}{nX}\sin\delta \quad (6)$$

$$Q_d = \frac{v_0 v}{nX}\cos\delta - \frac{v^2}{n^2 X} \quad (7)$$

Combining the load dynamics (1)–(4) with power balance equations at the load bus (6)–(7), the model can be written in the differential-algebraic (DA) for

$$\dot{x} = f(x, y, u) \quad (8)$$

$$0 = g(x, y, u) \quad (9)$$

where $x = [x_p \ x_q]^T$ is the dynamic state vector, $y = [v \ \delta]^T$, containing the voltage amplitude and phase at the load bus, is the algebraic state vector. The

[1]nominal power is the power consumed by the load in steady-state at 1 p.u. voltage

[2]note that the voltage on the primary side of the transformer is (v/n)

input vector is $u = [n\ k\ X]^T$. The tap ratio n is limited to then interval 0.8-1.2 p.u. with steps of 0.02 p.u. and the load shedding input is limited to the range 0-0.15 p.u., with steps of 0.05 p.u. The line reactance X, which normally would be considered as a parameter is here used as a (measurable) disturbance input that model the outage of a transmission line. With the value X=0.25 p.u., the impedance corresponds to two parallel transmission lines in operation and with X=0.5 p.u. to the case when one of them is faulty.

For the example system in Figure 1, functions f and g can be written as

$$f(x,y) = \begin{bmatrix} -\frac{x_p}{T_p} + P_0(v^{\alpha_s} - v^{\alpha_t}) \\ -\frac{x_q}{T_q} + Q_0(v^{\beta_s} - v^{\beta_t})) \end{bmatrix} \tag{10}$$

$$g(x,y,u) = \begin{bmatrix} \frac{v_0 v}{nX}\sin\delta + (1-k)(x_p + P_0 v^{\alpha_t}) \\ -\frac{v_0 v}{nX}\cos\delta + \frac{v^2}{n^2 X} - B_0 v^2 + (1-k)(x_q + Q_0 v^{\beta_t}) \end{bmatrix} \tag{11}$$

2.2 Primary Control Systems: OLTC Model

The conventional OLTC control is a simple incremental control with a time delay and a deadband. The size of the dead zone sets the tolerance for long term voltage deviations and the time delay is primarily intended for noise rejection. Detailed descriptions of OLTC control systems can be found in (Sauer and Pai, 1994).

The typical non-sequential OLTC control system can be modelled by the state graph of Fig. 2. The system remains in the state *wait* while the voltage deviation ($|v - v_r|$) is less than the function voltage (u_{function}). When the limit is exceeded, a transition to the state *count* occurs. Upon entering *count*, a timer is started and is kept running until either it reaches the delay time T_d, causing a transition to the state *action*, or if the voltage deviation becomes less than the reset voltage (u_{reset}), firing a transition to the state *wait* and reset of the timer. When entering the state *action*, a control pulse to operate the tap changer is given. After the mechanical delay time (T_m), the tap operation is completed. The control system then receives *ready* signal from the tap changer and returns to state *wait*. The time delay is tuned by the time delay constant T_{d0}. The actual time delay can then be either fixed ($T_d = T_{d0}$) or inversely proportional to the voltage deviation ($T_d \sim T_{d0}/|v - v_r|$).

For agreement with Sauer and Pai (1994), function and reset voltages have been chosen to be identical ($u_{\text{function}} = u_{\text{reset}} = DB/2$). Possible tap ratio change is typically $\pm 10 - 15$ % in steps of 0.6-2.5 %. Typical setting of delay times (T_d) is in the range 30-120 s, whilst the deadband (DB) is usually chosen slightly smaller than two tap steps. The mechanical delay time (T_m) is usually in the range 1-5 s.

Several models with different types of time delay are given by Sauer and Pai (1994). Here, we will restrict our attention to the case with constant delay time. Parameter data can be found in the Modelica code in Appendix A.

2.3 Simplified Primary Controller Representation

To allow simpler analysis of the system an additional simplification can be be made regarding the primary control system for the tap changer. Using a discrete time representation of the OLTC dynamics, the tap changer controller can be described by

$$n(k+1) = \begin{cases} n(k)+1 & \text{if } (v-v_r) < -DB/2 \text{ and } n(k) < n_{max} \\ n(k) & \text{if } -DB/2 < v < DB/2 \\ n(k)-1 & \text{if } (v-v_r) > DB/2 \text{ and } n(k) > n_{min} \end{cases} \qquad (12)$$

This model only gives a good approximation for constant delay time, and the sample time of the discrete-time system should then be chosen as $h = T_{d0} + T_m$. Figure 5 illustrates the effect of approximating the tap changer dynamics following a simulated line tripping (change of X from the initial value 0.25 p.u. to 0.5 p.u.). The response is essentially the same with the approximated version, although the first tap changer operation is carried out slightly earlier for the approximated version because of the periodic sampling. This approximation is commonly used by power systems engineers (Van Cutsem and Vournas, 1998).

3 Closed Loop System Simulation

Figure 3 shows the response of the system to a simulated line tripping (change of X from the initial value 0.25 p.u. to 0.5 p.u.) for three cases.

Firstly the open loop system as X given by (8)–(9) with n and k fixed at their default values 1 and 0, respectively. Following the fault, the voltage monotonically decreases until equation (9) becomes unsolvable at around time 316.8 s.

In the second simulation, the same disturbance is applied with the primary controls enables, the input n is then generated by the state-automata in Figure 2. The load shedding input k is still fixed at 0. Following the fault, a small voltage recovery is provided by each tap step made by the tap changer controller. However, the tap changer control is only effective until about simulation time 250 s. Thereafter, each new step actually has a negative effect on the load voltage. Equation (9) becomes unsolvable at around time 316.8 s.

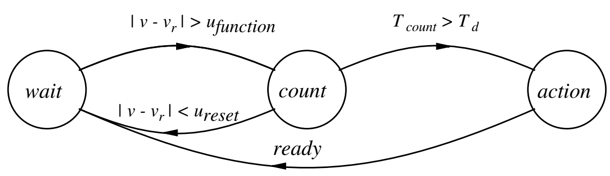


Figure 2: State graph illustrating function of a non-sequential OLTC control system.

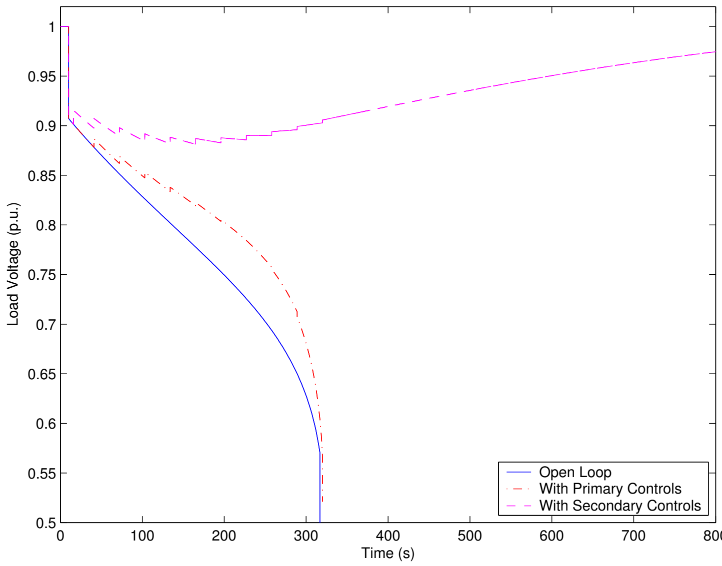


Figure 3: Simulation results for system without primary controls (solid line), with primary control applied (dash-dotted line) and with secondary (and primary) controls applied (dotted line).

In the third simulation, typical undervoltage load shedding relays have been applied as a secondary control layer (a detailed description can be found in Vu et al. (1995), and in the Modelica HTML documentation). Five percent of the load is disconnected after time delays of 1.5,3 and 6 s, at voltage 0.8, 0.82 and 0.92 p.u, respectively. The reason for the staggering of the time delay and voltage thresholds is to ensure proper selectivity when many loads are equipped with load shedding relays. The secondary control strategy executes a 5 % load shedding at time 16 s and the load voltage and the voltage appears to stabilize close to nominal voltage. Simulation for an extended time as shown in Figure 4, reveals that the system experiences a stable limit cycle originating in the interaction of the discrete tap changer dynamics and the continuous load dynamics.

4 Conclusion

A simple test case to illustrate load-voltage dynamics in power system has been illustrated by simulation and described in terms of equations for the complete model. The test system model contains a generator, a transmission line and an ideal transformer with its' associated control system. Furthermore, an approximate version of this control system has been described.

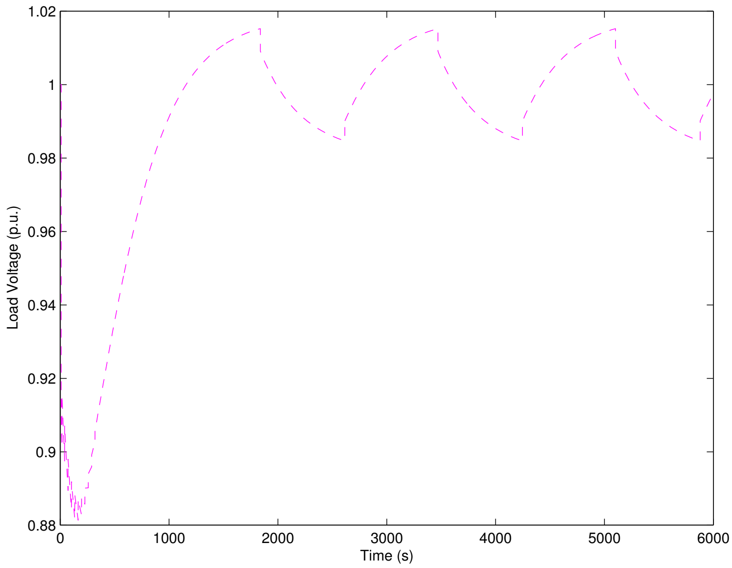


Figure 4: Simulation result with secondary control applied.

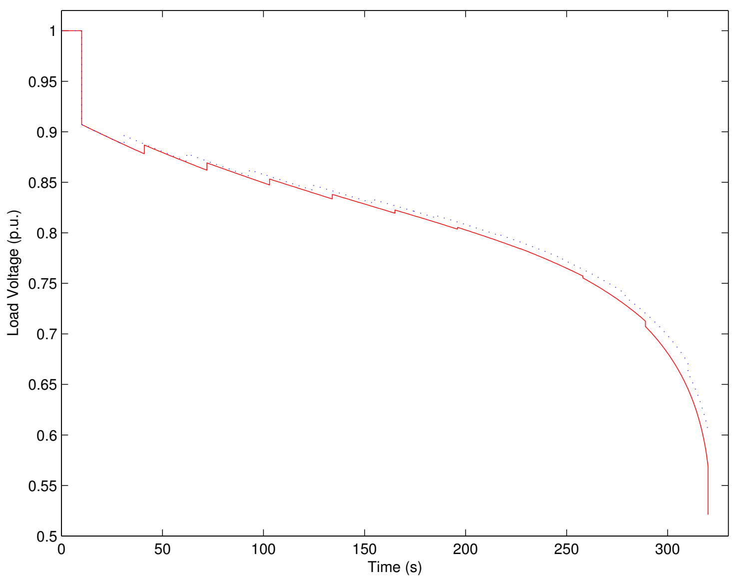


Figure 5: Illustration of the effect of approximating the tap change dynamics.

5 Acknowledgement

This work was partly funded by a grant from the Bundesamt für Bildung und Wissenschaft, Bern, Schweiz in the Control and Computation project (IST-2001-33520).

References

Glover, J. D. and Sarma, M. (1994). *Power System Analysis and Design*. PWS Publishing Company. ISBN 0-53493-960-0.

Karlsson, D. and Hill, D. J. (1994). Modelling and identification of nonlinear dynamic loads in power systems. *IEEE Transactions on Power Systems*, vol. 9, no. 1, pp. 157–63.

Sauer, P. W. and Pai, M. A. (1994). A comparison of discrete vs. continuous dynamic models of tap-changing-under-load transformers. In *Proceedings Bulk Power System Voltage Phenomena - III : Voltage Stability, Security and Control, Davos, Switzerland*.

Van Cutsem, T. and Vournas, C. (1998). *Voltage Stability of Electric Power Systems*. Power Electronics and Power Systems Series. Kluwer Academic Publishers. ISBN 0-7923-8139-4.

Vu, K. T., Liu, C.-C., Taylor, C. W. and Jimma, K. M. (1995). Voltage instability: mechanisms and control strategies. *Proceedings of the IEEE*, vol. 83, no. 11, pp. 1442–55.

A Modelica Models and Documentation

ABB_CC_Testcase.TwoNode

ABB_CC_Tes ABB_CC_Tes ABB_CC_Tes ABB_CC_Tes ABB_CC_Tes ABB_CC_Tes ABB_CC_Tes ABB_CC_Tes ABB_CC_Tes
ABB_CC_Tes ABB_CC_Tes

Information

The twonode package contiant the Modelica code for the 'mini

ABB_CC_Testcase.TwoNode.OpenLoop

ABB_CC_Tes

ABB_CC_Testcase.TwoNode.OpenLoop

Information

Model of the Open Loop mini test case built using the standard component library.

Modelica definition

```
model OpenLoop
  Components.Slack G1(V0=1.05504739284515);
  Components.Bus Bus1;
  Components.Varimp L1;
  Components.Impedance L2(X=0.5);
  Components.Bus Bus2;
  Components.Bus Bus3;
  ABB_CC_Testcase.Components.Load Load(
    P0=1,
    Tp=60,
    Tq=60,
    Q0=0.3);
  ABB_CC_Testcase.Components.Transformer T1(X=1e-9);
  input Integer LoadStep;
  input Integer TapStep;
  ABB_CC_Testcase.Components.FixCapacitor FixCapacitor1(B=0.2);
equation
  connect(Load.T, Bus3.T);
  //  L1.X = if time > 20 then Modelica.Constants.inf else 0.5;
  L1.X = if time > 10 then 1e5 else 0.5;
  Load.step = LoadStep;
  T1.tappos = TapStep;

  connect(G1.T, Bus1.T);
```

```
  connect(L2.T1, G1.T);
  connect(L1.T2, G1.T);
  connect(Bus2.T, T1.T1);
  connect(T1.T2, Bus3.T);
  connect(FixCapacitor1.T, Bus3.T);
  connect(L1.T1, Bus2.T);
  connect(L2.T2, Bus2.T);
end OpenLoop;
```

ABB_CC_Testcase.TwoNode.OpenLoopEq

ABB_CC_Tes

Information

```
Model of the Open Loop mini test case built using direct mathematical modelling.
This model is equivalent to OpenLoop.
```

Parameters

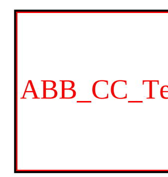
Name	Default	Description
v0	1.05504739284515	
P0	1	
Q0	0.3	
B0	0.2	
as	0	
at	2	
bs	0	
bt	2	
Tp	60	
Tq	60	
faultTime	10	

Modelica definition

```
model OpenLoopEq
  // parameters
  parameter Real v0=1.05504739284515;
  parameter Real P0=1;
  parameter Real Q0=0.3;
  parameter Real B0=0.2;
  parameter Real as=0;
  parameter Real at=2;
  parameter Real bs=0;
  parameter Real bt=2;
  parameter Real Tp=60;
  parameter Real Tq=60;
  parameter Real faultTime=10;

  output Real x[2]={xp,xq};
  output Real y[2](start={1,0}) = {v,delta};
```

```
  input Real u[3]={n,k,X};
  // variables
  Real xp;
  Real xq;
  Real v(start=1);
  Real delta;
  Real k;
  Real n;
  Real X;
equation
  // vector versions
  {der(xp),der(xq)} = {(-xp/Tp + P0*(v^as - v^at)),(-xq/Tq + Q0*(v^bs - v^bt))}
    ;

  {0,0} = {v*sin(delta)*v0/(X*n) + (1 - k)*(xp/Tp + P0*v^at),-(v0*v*cos(delta))
    /(n*X) + v^2/(n^2*X) - B0*v^2 + (1 - k)*(xq/Tq + Q0*v^bt)};
initial equation
  der(xp) = 0;
  der(xq) = 0;
end OpenLoopEq;
```

ABB_CC_Testcase.TwoNode.PrimaryController



Primary Control System

Information

```
This model models the primary control systems of the
four node power system.
It contains a tap changer controller for transformer T1
modelled as a state machine and an automatic voltage regulator
with filed voltage limiter modelled by a state machina and a linear
control law.
The special equations activated by the flag SimulinkSafe
flag are jus fixes that must be applied to make the model simulable with
Simulink. If you remodel this system with other tools,
use the original equations (assume that SimulinkSafe=false)
```

Parameters

Name	Default	Description
idle0	false	
wait0	true	
action0	false	
tappos0	0	
timer0	0	
up_limit0	false	
TapDelay	30	
MechDelay	1	
TapDB	0.01*3	

MaxTap	10	
MinTap	-10	

Modelica definition

```
model PrimaryController "Primary Control System"
  parameter Boolean idle0=false;
  parameter Boolean wait0=true;
  parameter Boolean action0=false;
  parameter Integer tappos0=0;
  parameter Real timer0=0;
  parameter Boolean up_limit0=false;

  input Real TapVref(start=1);
  input Real v(start=1);

  Boolean idle(start=idle0, fixed=true);
  Boolean wait(start=wait0, fixed=true);
  Boolean action(start=action0, fixed=true);
  Integer tappos(start=tappos0, fixed=true);
  Real timer(start=timer0, fixed=true);

  Boolean toohigh;
  Boolean toolow;

  output Integer T1tappos;

  parameter Real TapDelay=30;
  parameter Real MechDelay=1;
  parameter Real TapDB=0.01*3;
  parameter Real MaxTap=10;
  parameter Real MinTap=-10;

equation
  T1tappos = pre(tappos);
  // tap changer control - state automata !
  toohigh = (v - TapVref) > TapDB/2;
  toolow = (v - TapVref) < -TapDB/2;

  idle = (pre(idle) or pre(wait)) and not (toohigh or toolow) or (pre(action)
     and ((time - timer) > TapDelay + MechDelay));
  wait = (pre(idle) and (toohigh or toolow)) or pre(wait) and ((toolow or
    toohigh) and ((time - pre(timer)) < TapDelay));
  action = (pre(wait) and (time - timer > TapDelay)) or pre(action) and ((time
    - timer) < TapDelay + MechDelay);
  when wait and not pre(wait) and not initial() then
    timer = time;
  end when;
  when pre(action) and not action then
    if toolow and (pre(tappos) < MaxTap) then
      tappos = pre(tappos) + 1;
    elseif toohigh and (pre(T1tappos) > MinTap) then
      tappos = pre(tappos) - 1;
    else
      tappos = pre(tappos);
    end if;
  end when;
end PrimaryController;
```

ABB_CC_Testcase.TwoNode.PrimaryControlledSystem

ABB_CC_Tes

ABB_CC_Testcase.TwoNode.PrimaryControlledSystem

Information

Connection of OpenLoop and Primary control systems

Parameters

Name	Default	Description
faultTime	10	
tapstepsize	0.02	
loadstepsize	0.05	

Modelica definition

```
model PrimaryControlledSystem
  ABB_CC_Testcase.TwoNode.OpenLoopEq System;
  ABB_CC_Testcase.TwoNode.PrimaryController PrimCon;
  parameter Real faultTime=10;
  parameter Real tapstepsize=0.02;
  parameter Real loadstepsize=0.05;

  input Real LoadStep;
equation
  System.u = {1 + tapstepsize*PrimCon.T1tappos,LoadStep*loadstepsize,if time >
    faultTime then 0.5 else 0.25};
  PrimCon.v = System.v;
  PrimCon.TapVref = 1;
end PrimaryControlledSystem;
```

ABB_CC_Testcase.TwoNode.ClosedLoop

ABB_CC_Tes

Information

Connection of OpenLoop and Primary control systems with no secondary control.

Modelica definition

```
model ClosedLoop
  ABB_CC_Testcase.TwoNode.PrimaryControlledSystem System;
equation
  System.LoadStep = 0;
end ClosedLoop;
```

ABB_CC_Testcase.TwoNode.ClosedLoop2

ABB_CC_Tes

ABB_CC_Testcase.TwoNode.ClosedLoop2

Information

```
Connection of OpenLoop and Primary control systems with secondary control using
undervoltage load shedding relays.
```

Modelica definition

```
model ClosedLoop2
  ShedRelay Relay1(
    Threshold=.8,
    DelayTime=1.5,
    ShedAmount=.05);
  ShedRelay Relay2(
    Threshold=.82,
    DelayTime=3,
    ShedAmount=.05);
  ShedRelay Relay3(
    DelayTime=6,
    ShedAmount=0.05,
    Threshold=.92);
equation
  Relay1.u = System.System.v;
  Relay2.u = System.System.v;
  Relay3.u = System.System.v;
  System.System.k = Relay1.y + Relay2.y + Relay3.y;
public
  ABB_CC_Testcase.TwoNode.PrimaryControlledSystem System;
end ClosedLoop2;
```

ABB_CC_Testcase.TwoNode.NoControl

ABB_CC_Tes

TwoNode TestCase without primary controls

ABB_CC_Testcase.TwoNode.NoControl

Information

```
This is a model of the twonode testcase, where no primary control systems has been
connected.
Instead the tap ratio is fixed at 1 p.u. and the load shedding input at 0 p.u.

A fault on the the line is modelled of the change of the line impedance from 0.25 to
0.5 p.u. at
time 10 s.
```

Parameters

Name	Default	Description
faultTime	10	

Modelica definition

```
model NoControl "TwoNode TestCase without primary controls"
  parameter Real faultTime=10;
  ABB_CC_Testcase.TwoNode.OpenLoopEq System;
equation
  System.u = {1,0,if time > faultTime then 0.5 else 0.25};
end NoControl;
```

ABB_CC_Testcase.TwoNode.ShedRelay

ABB_CC_Tes

ABB_CC_Testcase.TwoNode.ShedRelay

Information

Standard load shedding relay.

Parameters

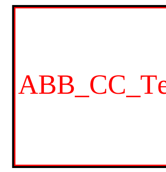
Name	Default	Description
Threshold	0.95	
DelayTime	1	
ShedAmount	0.05	

Modelica definition

```
class ShedRelay
  extends Modelica.Blocks.Interfaces.SISO;
  parameter Real Threshold=0.95;
  parameter Real DelayTime=1;

  parameter Real ShedAmount=0.05;
  discrete Real timerstart(start=-1, fixed=true);

  ModelicaAdditions.PetriNets.Transition T1;
  ModelicaAdditions.PetriNets.Transition T2;
  ModelicaAdditions.PetriNets.Place12 delay;
  ModelicaAdditions.PetriNets.Transition T3;
  ModelicaAdditions.PetriNets.Place21 wait(initialState=true);
equation
  connect(delay.inTransition, T1.outTransition);
  connect(T3.inTransition, delay.outTransition1);
  connect(delay.outTransition2, T2.inTransition);
  connect(T1.inTransition, wait.outTransition);
  connect(T2.outTransition, wait.inTransition2);
  connect(T3.outTransition, wait.inTransition1);
```

```
  T1.condition = (u < Threshold) and (y < ShedAmount/10);
  T2.condition = (u > Threshold);
  T3.condition = time > timerstart + DelayTime;

  when delay.state then
    timerstart = time;
  end when;

  when T3.condition and delay.state then
    outPort.signal[1] = ShedAmount;
  end when;
initial equation
  wait.state = true;
  delay.state = false;
  y = 0;
end ShedRelay;
```

ABB_CC_Testcase.TwoNode.SimplePrimaryController



Primary Control System

Information

```
This model models the primary control systems of the
four node power system using a simplified discrete-time approximation.
```

Parameters

Name	Default	Description
tappos0	0	
up_limit0	false	
TapDelay	30	
MechDelay	1	
TapDB	0.01*3	
MaxTap	10	
MinTap	-10	

Modelica definition

```
model SimplePrimaryController "Primary Control System"
  parameter Integer tappos0=0;
  parameter Boolean up_limit0=false;

  input Real TapVref(start=1);
  input Real v(start=1);

  Integer tappos(start=tappos0, fixed=true);

  Boolean toohigh;
  Boolean toolow;
```

```
  output Integer T1tappos;

  parameter Real TapDelay=30;
  parameter Real MechDelay=1;
  parameter Real TapDB=0.01*3;
  parameter Real MaxTap=10;
  parameter Real MinTap=-10;

equation
  when sample(0, TapDelay + MechDelay) then
    // tap changer control - simple discrete implementation
    toohigh = (v - TapVref) > TapDB/2;
    toolow = (v - TapVref) < -TapDB/2;
    if toolow and (pre(tappos) < MaxTap) then
      tappos = pre(tappos) + 1;
    elseif toohigh and (pre(T1tappos) > MinTap) then
      tappos = pre(tappos) - 1;
    else
      tappos = pre(tappos);
    end if;
  end when;
  T1tappos = pre(tappos);
end SimplePrimaryController;
```

ABB_CC_Testcase.TwoNode.SimplePrimaryControlledSystem

ABB_CC_Tes

ABB_CC_Testcase.TwoNode.SimplePrimaryControlledSystem

Parameters

Name	Default	Description
faultTime	10	
tapstepsize	0.02	
loadstepsize	0.05	

Modelica definition

```
model SimplePrimaryControlledSystem
  ABB_CC_Testcase.TwoNode.OpenLoopEq System;
  ABB_CC_Testcase.TwoNode.SimplePrimaryController PrimCon;
  parameter Real faultTime=10;
  parameter Real tapstepsize=0.02;
  parameter Real loadstepsize=0.05;

  input Real LoadStep;
equation
  System.u = {1 + tapstepsize*PrimCon.T1tappos,LoadStep*loadstepsize,if time >
    faultTime then 0.5 else 0.25};
  PrimCon.v = System.v;
  PrimCon.TapVref = 1;
end SimplePrimaryControlledSystem;
```

ABB_CC_Testcase.TwoNode.ComparePrimaryControls

ABB_CC_Tes

Modelica definition

```
model ComparePrimaryControls
  ABB_CC_Testcase.TwoNode.SimplePrimaryControlledSystem SimpleSystem;
  ABB_CC_Testcase.TwoNode.PrimaryControlledSystem System;
equation
  System.LoadStep = 0;
  SimpleSystem.LoadStep = 0;
end ComparePrimaryControls;
```

HTML-documentation generated by Dymola Thu Mar 27 10:32:15 2003 .