Kfz-Bordnetz

Einfache Teilmodelle in VHDL-AMS

Hella KG Hueck & Co.

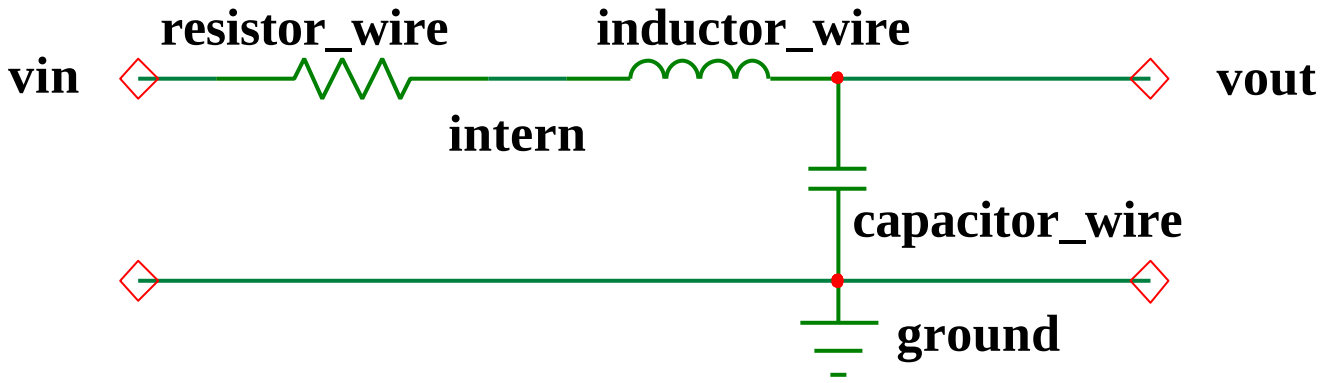
Ewald Hessel

Dipl.-Ing.
Zentrale Elektronik
Simulations- und Softwaretechnik

Hella KG Hueck & Co.
Standort: Werk 2
Beckumer Str. 130
59555 Lippstadt
Telefon 02941 / 38-8572
Telefax 02941 / 30-47 8572
e-mail: Ewald.Hessel@hella.de

Elektrisches Verhaltensmodell einer einfachen Leitung im Kfz

Elektrisches Ersatzbild



Definition der ENTITY 'capacitor_wire' und des Verhaltens (ARCHITECTURE)

```
-- Function : model of a simple capacitor within a
-- a model of a line
--
-- Author: Ursula Kemper, Hella KG
-- Date: Jan. 2001
-------------------------------------------------------------------------
--
LIBRARY IEEE;
USE IEEE.math_real.ALL;
USE WORK.energy_systems.ALL;
USE WORK.electrical_systems.ALL;

ENTITY capacitor_wire IS

  GENERIC (l: REAL := 1.0;
           d: REAL := 2.0;
           a: REAL := 1.0);
   PORT (TERMINAL p,
                  m :     electrical);

END ENTITY capacitor_wire;
```

```
ARCHITECTURE msr_simple OF capacitor_wire IS

 CONSTANT h0 : REAL := a+d/2.0;      -- equation simplifier

 CONSTANT cnom :

REAL := 0.241*(1.0e-12) * l / (log((2.0*h0/d) *(1.0+sqrt(1.0-
1.0/(2.0*h0/d)**2.0))));  -- calculation of capacitor

 QUANTITY vc across ic through p TO m;

-- Quelle:

-- Meinke/Gundlach : Taschenbuch der Hochfrequenztechnik


BEGIN


  ic == cnom * vc'dot;


END ARCHITECTURE simplewire;
```

Definition der ENTITY ‘resistor_wire‘ und des Verhaltens (ARCHITECTURE)

```
-- Function : model of a simple resistor within
-- a model of a line
--
-- Author: Ursula Kemper, Hella KG
-- Date: Jan. 2001
-------------------------------------------------------------------------
--
USE WORK.energy_systems.ALL;
USE WORK.electrical_systems.ALL;

ENTITY resistor _wire IS

GENERIC (     l:    REAL := 1.0;       -- [m] length
              d:    REAL := 2.0;       --[mm] diameter
            rho:    REAL := 0.0172);

PORT (TERMINAL p, m : electrical);

END ENTITY resistor_wire;
```

```
ARCHITECTURE msr_simple OF resistor_wire IS

CONSTANT  pi: REAL := 3.1415936;
CONSTANT rnom : REAL := (rho*l*4.0)/(pi*d**2.0);
 -- resistor value depending on lenghth, diameter,
 -- specific resistance
-- Quelle:
-- Meinke/Gundlach : Taschenbuch der Hochfrequenztechnik

 QUANTITY vr across ir through p TO m;

BEGIN  -- ARCHITECTURE msr_simple

  ir == vr / rnom;

END ARCHITECTURE msr_simple;
```

Definition der ENTITY ‘inductor_wire‘ und des Verhaltens (ARCHITECTURE)

```
-- Function : model of a simple inductor within
-- a model of line
--
-- Author: Ursula Kemper, Hella KG
-- Date: Jan. 2001
-------------------------------------------------------------------------
--
LIBRARY IEEE;
USE work.ai_standard.ALL;
USE IEEE.math_real.ALL;
USE work.energy_systems.ALL;
USE work.electrical_systems.ALL;

ENTITY inductor_wire IS

   GENERIC ( l: REAL := 1.0;
                  a                    : REAL := 1.0;
                  mu       : REAL := 1.0;
                  d                    : REAL := 2.0;
                  k3       : REAL := 0.009);
   PORT ( TERMINAL p,
               m : electrical);
```

```
ARCHITECTURE msr_simple OF inductor_wire IS

 CONSTANT  lnom : REAL := 2.0*l*100.0*(1.0e-6) *
(log(4.0*a/d)+mu*k3)*(1.0e-3);  -- calculation of inductivity


-- Quelle:

-- Meinke/Gundlach : Taschenbuch der Hochfrequenztechnik


QUANTITY vl across il through p TO m;


BEGIN

  vl == lnom * il'dot;

END ARCHITECTURE msr_simple;
```

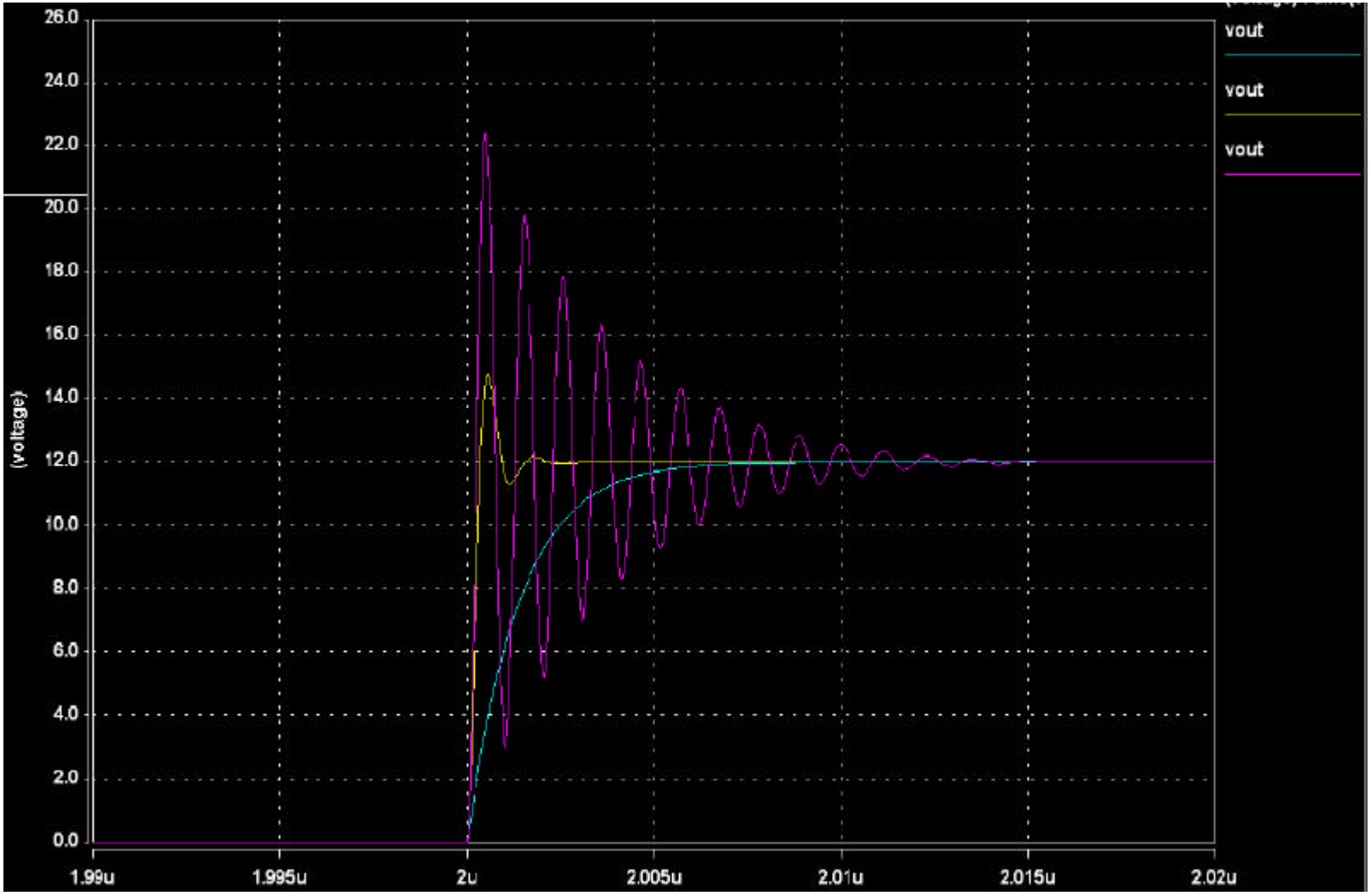
Definition der ENTITY ‘wire‘ und des Verhaltens (ARCHITECTURE)

```
-- Function : model of a simple line in automobile
--
-- Author: Ursula Kemper, Hella KG
-- Date: Jan. 2001
-------------------------------------------------------------------------------
Library IEEE;
USE work.electrical_systems.ALL;
use ieee.math_real.all;
ENTITY wire IS
   GENERIC (l  : real;          -- [m] wire length
                  d : real;          -- [mm] wire diameter
                  a : real ;           -- [mm] distance
                  rho : real;       -- specific density
                  mu : real;        -- permeability
                  k3 : real);       -- experimental value

   PORT (
   TERMINAL vin  : electrical;         -- input
   TERMINAL vout : electrical);       -- output

END ENTITY wire;
```

```
ARCHITECTURE msr_simple OF wire IS
TERMINAL  intern : electrical;     -- internal node

BEGIN  -- ARCHITECTURE msr_simple
 r1 : ENTITY work.resistor_wire
      GENERIC MAP (l => l, d => d, rho => rho)
      PORT MAP       (p => vin, m  => intern);

 l1 : ENTITY work.inductor_wire
      GENERIC MAP (l => l, d => d, a => a, mu => mu,  k3 => k3)
      PORT MAP (p  => intern, m  => vout);

 c1 : ENTITY work.capacitor_wire
      GENERIC MAP (l => l,  d => d,  a => a)
      PORT MAP ( p => vout, m  => ground);

END ARCHITECTURE msr_simple;
```

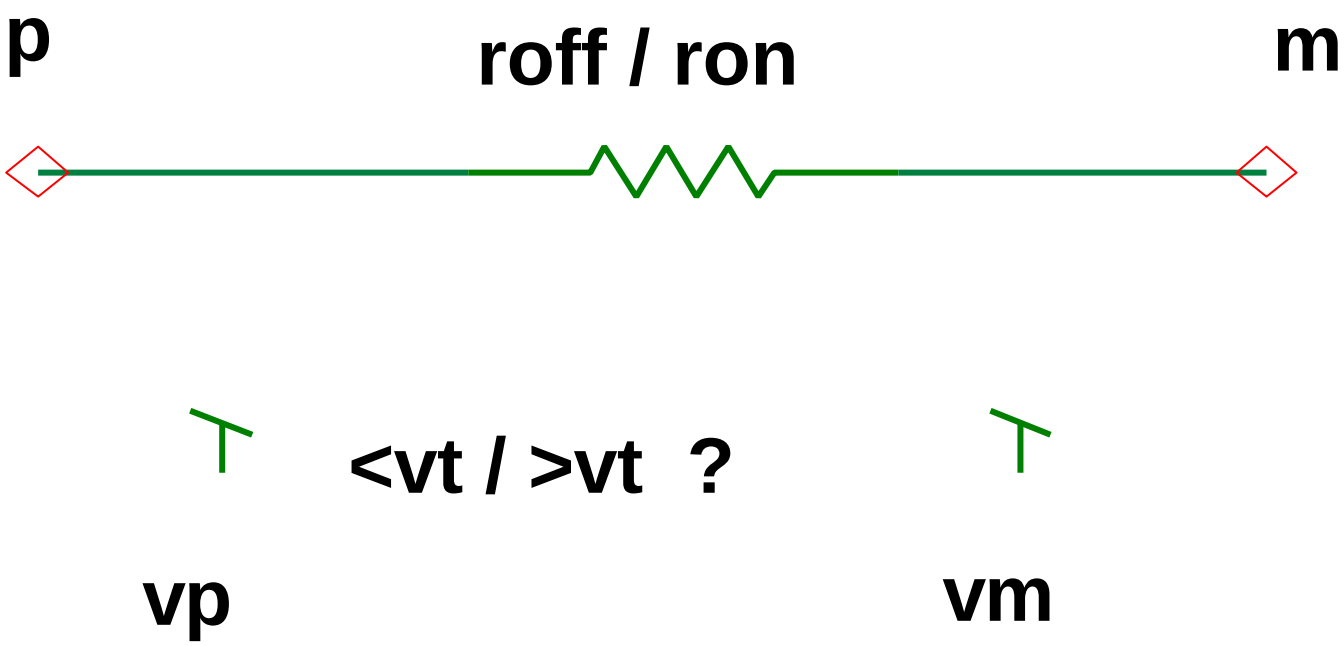
Einschaltverhalten vout=f(time) bei verschiedenen Lasten (100,1k,10k Ohm)

vout / V



time/sec

Elektrisches Verhaltensmodell eines Schalters als gesteuerter Widerstand



Definition der ENTITY ‘switch_simple‘ und des Verhaltens (ARCHITECTURE)

```
-- Function : model of a simple switch in automobile
--
-- Author: Ursula Kemper, Hella KG
-- Date: Jan. 2001
-------------------------------------------------------------------------------
LIBRARY ieee;
USE WORK.energy_systems.ALL;
USE WORK.electrical_systems.ALL;

ENTITY switch_simple IS
GENERIC (vt : real := 0.0; -- threshold voltage
  ron : real := 1.0;          -- switch "on" resistance
  roff : real := 1.0e12);     -- swicth "off" resistance

PORT (TERMINAL
  p, m : electrical;  -- pos., neg. connection of switch

TERMINAL
  vp, vm : electrical);  -- connection of control voltage

 END switch_simple;
```

```
ARCHITECTURE msr_simple OF switch_simple IS
  QUANTITY v across i through p TO m;
  QUANTITY vcontrol across vp TO vm;
BEGIN  -- msr_simple
    IF vcontrol'ABOVE(vt) USE
       i == v/ron;
     ELSE
       i == v/roff;
     END USE;
break ON vcontrol'ABOVE(vt);

END ARCHITECTURE msr_simple;
```

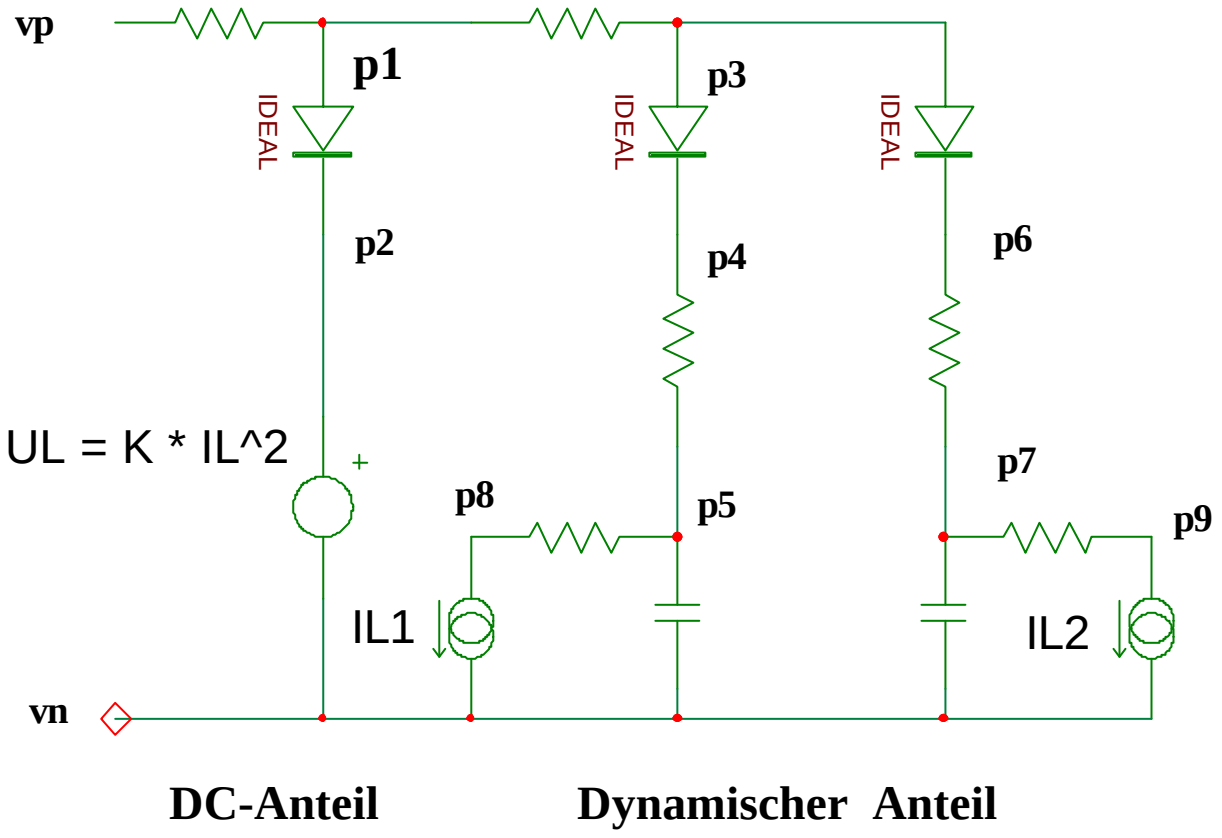
Definition der ENTITY ‘diode_simple‘ und des Verhaltens (ARCHITECTURE)

```
-- Function : model of a simple switch in automobile
--
-- Author: Ewald Hessel, Hella KG
-- Date: Jan. 2001
------------------------------------------------------------------------
LIBRARY IEEE;
USE IEEE.MATH_REAL.ALL;
USE WORK.energy_systems.ALL;
USE WORK.electrical_systems.ALL;

ENTITY diode_simple IS

GENERIC (i0   : REAL := 1.0e-9;  -- [A] leakage current
         ut   : REAL := 25.0e-3); -- [V] thermal voltage
         ron: REAL :=  1.0e-6; -- [Ohm]
         roff: REAL := 1.0e15; -- [Ohm]

PORT (TERMINAL p, m :electrical);

END ENTITY diode;
```

```
ARCHITECTURE msr_simple OF diode_simple IS

QUANTITY ud across id through p TO m;

BEGIN  -- ARCHITECTURE simple

if id'ABOVE(0.0) USE
    UD == id * ron;
ELSE UD == id * roff;
END USE;
BREAK on id'ABOVE(0.0);

--  id == I0*(exp(ud/ut)-1.0);

END ARCHITECTURE simple;
```

Elektrisches Verhaltensmodell einer Glühlampe

Elektrisches Ersatzschaltbild



Definition der ENTITY ‘lampe‘ und des Verhaltens (ARCHITECTURE) ,Teil 1

```
-- Function : dynamical elctrical behavioral of a lamp
--
-- Author: E. Hessel, Hella KG
-- Date: 20.02. 2001
--------------------------------------------------------------------------------
LIBRARY ieee;
USE work.electrical_systems.all;
use ieee.math_real.all;

ENTITY lampe IS

 GENERIC (un  : real := 13.5;  -- [Volt] Nennspannung

                pn  : real := 25.0); -- [Watt] Nennleistung

PORT ( TERMINAL vp,vn: electrical);

END ENTITY lampe;
```

```
ARCHITECTURE msr_simple OF lampe IS

TERMINAL  p1,p2,p3,p4,p5,p6,p7,p8,p9 : electrical;

CONSTANT  fi1     : REAL := 15.0;
CONSTANT  fi2  : REAL := 3.0;
CONSTANT  t1   : REAL := 2.0e-3;
CONSTANT  t2   : REAL := 30.0e-3;
CONSTANT  t3   : REAL := 3.0;
CONSTANT  rn   : REAL := (un**2)/pn;
CONSTANT  r_r1 : REAL := rn/( fi1-1.0);
CONSTANT  w_c1 : REAL := t1/(rn/(fi1-1.0));
CONSTANT  w_r2 : REAL := rn/( fi2-1.0);
CONSTANT  w_c2 : REAL := t2 /(rn/(fi2-1.0));
CONSTANT  ki1  : REAL := t1 /(rn/(fi1-1.0))/t3;
CONSTANT  ki2  : REAL := t2 /(rn/(fi2-1.0))/t3;
```

Definition des Verhaltens (ARCHITECTURE)
Teil 2

```
QUANTITY ul ACROSS il THROUGH p2 TO vn;
QUANTITY ud1 ACROSS p4 TO p3;
QUANTITY ud2 ACROSS p6 TO p3;
QUANTITY il1 THROUGH p5 TO p8;
QUANTITY il2 THROUGH p7 TO p9;
```

```
BEGIN
ul == il**2 *((un**3)/(pn**2));
il1 == ud1 * ki1;
il2 == ud2 * ki2;
```

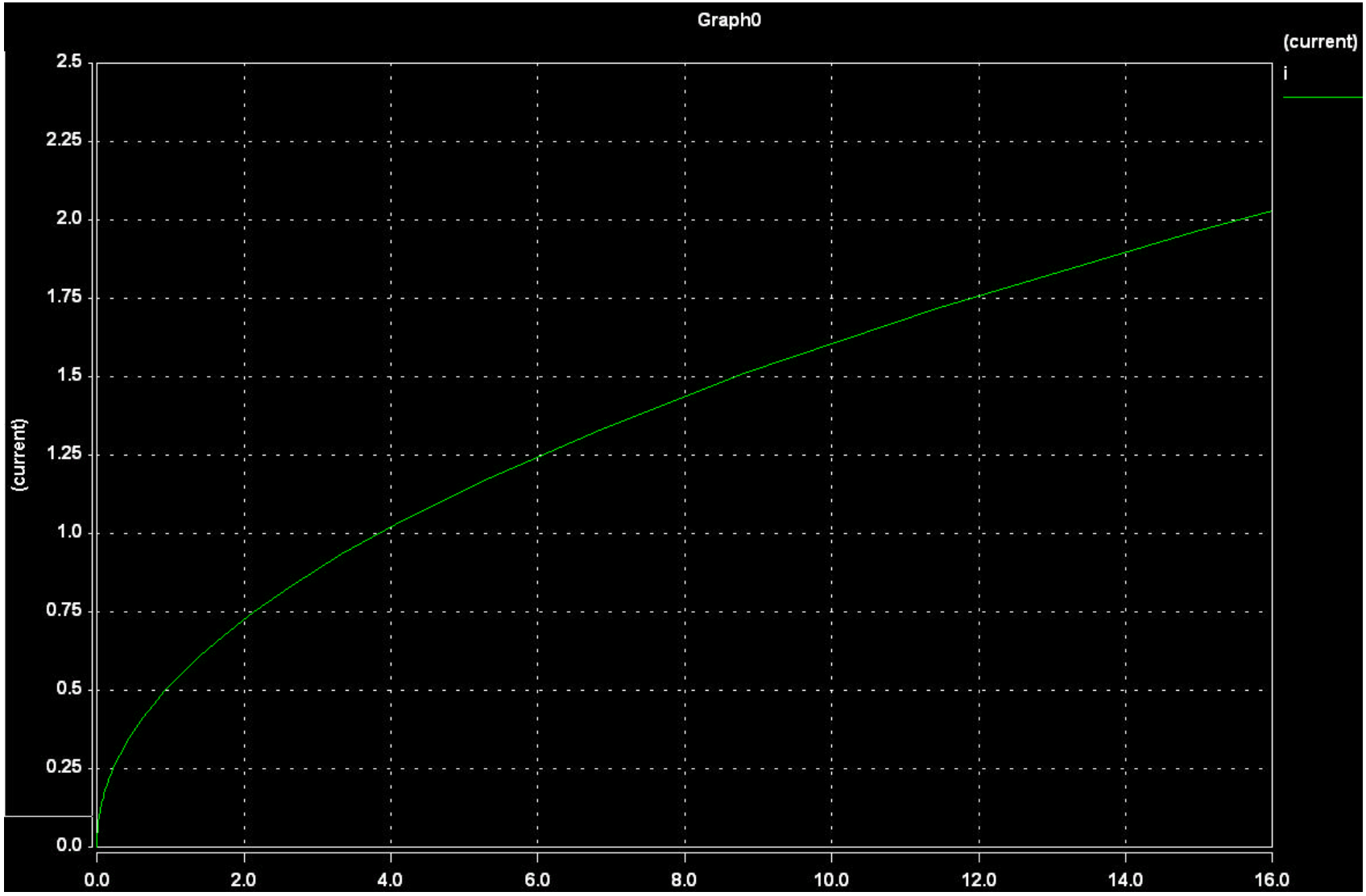
```
r3:  ENTITY work.resistor
     GENERIC MAP  (rnom => 1.0e-4)
     PORT MAP     (p  => p8,
                   m => vn);
r4 :  ENTITY work.resistor
     GENERIC MAP  (rnom => 1.0e-4)
     PORT MAP     ( p => p9,
                   m => vn);
rv1 : ENTITY work.resistor
     GENERIC MAP (rnom => 1.0e-4)
     PORT MAP       (p => vp,
                     m => p1);
rv2 : ENTITY work.resistor
     GENERIC MAP (rnom => 1.0e-4)
     PORT MAP       (p => p1,
                   m => p3);
```

Definition des Verhaltens (ARCHITECTURE)
Teil 3

```
dl:   ENTITY work.diode
      GENERIC MAP ( i0 => 1.0e-9,
                    ut => 25.0e-3)
      PORT MAP    ( p => p1,
                    m => p2);
d1 :  ENTITY work.diode
      GENERIC MAP ( i0 => 1.0e-9,
                    ut => 25.0e-3)
      PORT MAP    ( p => p3,
                    m => p4);
r1 :  ENTITY work.resistor
      GENERIC MAP (rnom => w_r1 )
      PORT MAP    (p   => p4,
                   m   => p5);
c1 : ENTITY work.capacitor
     GENERIC MAP  (cnom => w_c1)
     PORT MAP     (p   => p5,
                   m    => vn);
```

```
d2 : ENTITY work.diode
     GENERIC MAP  (i0   => 1.0e-9,
                   ut  => 25.0e-3)
     PORT MAP     ( p   => p3,
                    m   => p6);
r2 : ENTITY work.resistor
     GENERIC MAP    rnom => w_r2)
     PORT MAP       (p  => p6,
                     m  => p7);
c2 : ENTITY work.capacitor
     GENERIC MAP    (cnom => w_c2)
     PORT MAP       (p    => p7,
                     m    => vn);

END ARCHITECTURE structural;
```

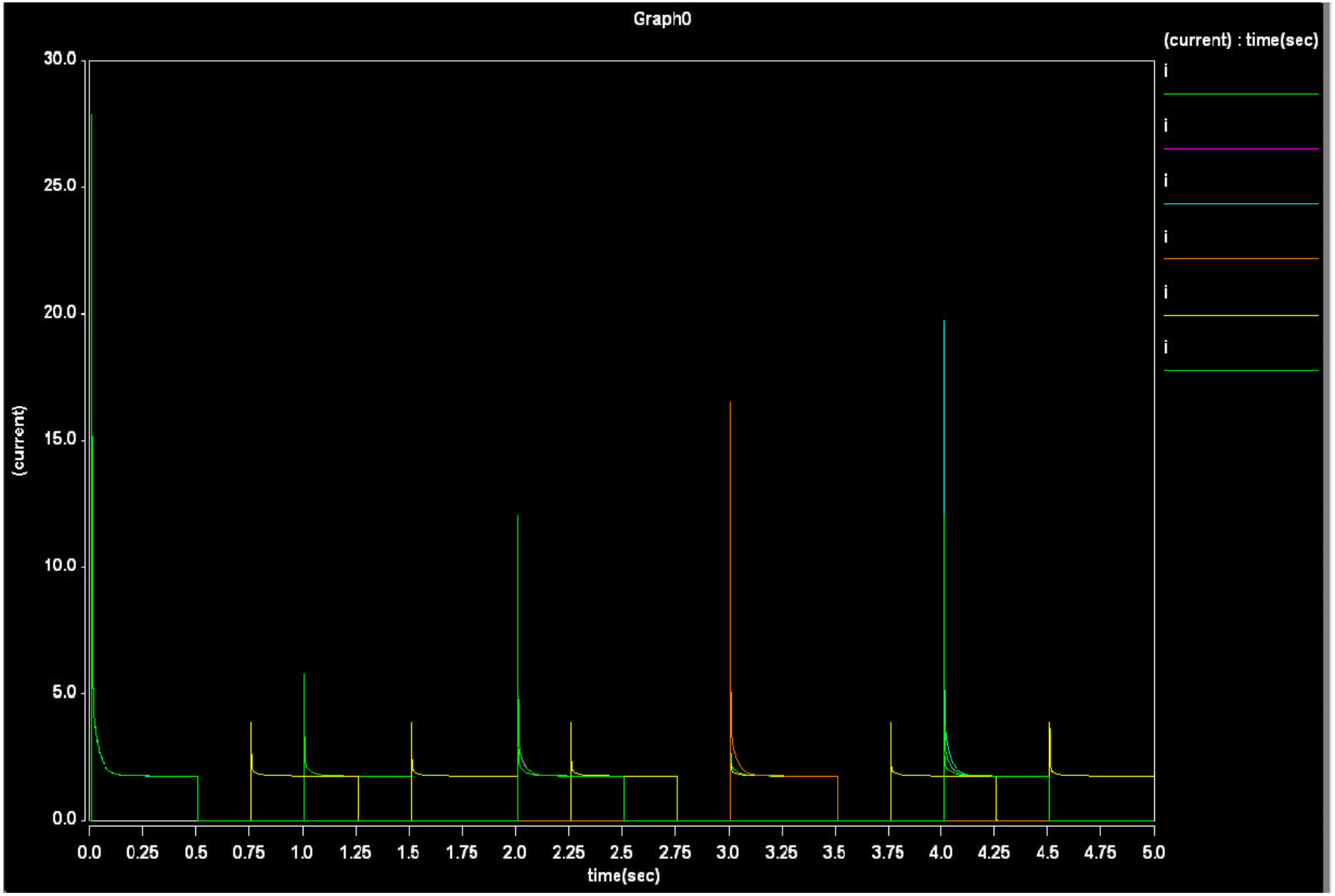
DC-Verhalten IL=f(UL)



IL / A

UL / V

Schaltverhalten IL=f(time)



IL / A

time/sec

Weiteres Vorgehen

-*Vervollständigung der Modellstruktur des Bordnetzes*

-*Ergänzung um weitere Module*

-*Verfeinerung der Modelle*

- *Optimierung hinsichtlich Stabilität und Performance*

-*Test der Gesamt-Modells auf unterschiedlichen Simulatoren*

-*Individuelle weitere Konkretisierung/Modellierung*

-*Erfahrungsbericht an Toolhersteller*