Journal of ELECTRICAL ENGINEERING, VOL. 57, NO. 5, 2006, 268–275

DISTANCE DIGITAL RELAY MODEL DEVELOPED IN ATP "FOREIGN MODEL" AND C++

František Janíček — Martin Mucha *

This paper presents a new approach in the problem of digital relay development. An interactive model in the simulation program of power system, *eg*, ATP-EMTP, could properly contribute to the improvement of digital relays. This model is not available and the code for this model has to be created in ATP. There is an internal programming language MODELS in ATP which allows creation of programs themselves but because of its simplicity it is not sufficient for developing sophisticated models as in object oriented programming language. This deficit is compensated for by allowing a suitable interconnection between the ATP computing core and external model developed in another programming language. For this reason, the object oriented programming language C++ was chosen for relay model development. The paper describes the way of linking and compiling of the relay model with ATP. Next, a multifunctional digital relay developed model is introduced with its functional elements, used algorithms and tripping characteristics. Simulation results are presented and correct tripping action is verified at the end.

K e y w o r d s: digital relay model, foreign model and C++, linking of foreign model, EMTP-ATP compiling, discrete Fourier transform

1 INTRODUCTION

Computer simulation of an interactive protection system in general includes modelling of the power system, modelling of the protection relay itself and of the dynamic interaction between the power system and the digital relay. This type of simulation is useful especially for preparatory testing of new protection relay algorithms, co-ordinated protection systems and for evaluating the relay performance.

For modelling the power system, digital relay and mutual interactions between them we can choose from among more options. First, an ATP/EMTP [4, 5], [7] code can be used to simulate the transient electromagnetic phenomenon in the power system, and TACS transient analysis to model the relay, or to use the programming language MODELS [6] of the simulating program ATP, which is an improvement of TACS. The main advantage of this approach is in simple interfacing between the power system and the digital relay model because the TACS and MODELS are parts of ATP/EMTP.

Next, one can use a model of the power system created in ATP and relay model developed in Matlab interconnected with ATP by an "interaction buffer" set for mutual exchange of data. It is a direct connection of the computation engine of Matlab to ATP.

The third option is a model of the power system created in Matlab/SimPower Systems, and the relay model is developed in Matlab/Simulink. Interfacing of models is easily attained because both are in Matlab.

Besides the mentioned advantages, single designs have also their limitations. In the first option the development of sophisticated relay models is considerably limited by the low flexibility of TACS and programmability of MODELS language. The second choice suffers from an excessive simulation time and the use of two programs causes a lack of integrity. In the last option, the simulation is slow when modelling large power systems in Matlab.

This paper describes a highly efficient approach to simulating the interactive protection system. In this case the power system is modelled in ATP and the digital relay model is developed in C++ programming language, which allows to program a highly sophisticated relay, consecutively linked and compiled with the source code of ATP. Software package MinGW32 [8] is used to link the relay source code with ATP and next to compile. Mutual interaction of the two models is then accomplished in MODELS language by using the submodel "foreign model", which ensures exchange of data between the two models by four data arrays.

2 LINKING AND COMPILING PROCEDURE OF ATP

Software package MinGW32 includes appropriate tools to link of the relay source code with ATP source code and to compile them. MinGW32 is a package of programs running under Windows. By *gcc.exe, g77.exe* and *make.exe* programs the fortran compiler will create the object file from FORTRAN ATP written in FORTRAN language and C++ compiler will create the object file from the relay source code written in C++ language. These will be subsequently linked with other object files and libraries and compiled, which will create a new simulation ATP program in file *tpbig.exe* containing a digital relay model in "foreign model". Two variables, pointing to MingW32 and source code files, in Windows

* Slovak University of Technology Bratislava, Faculty of Electrical engineering and Information Technology, Department of Electrical Power Engineering, Ilkovičova 3, 812 19 Bratislava 1, Slovakia, E-mail: frantisek.janicek@stuba.sk, martin.mucha@stuba.sk

ISSN 1335-3632

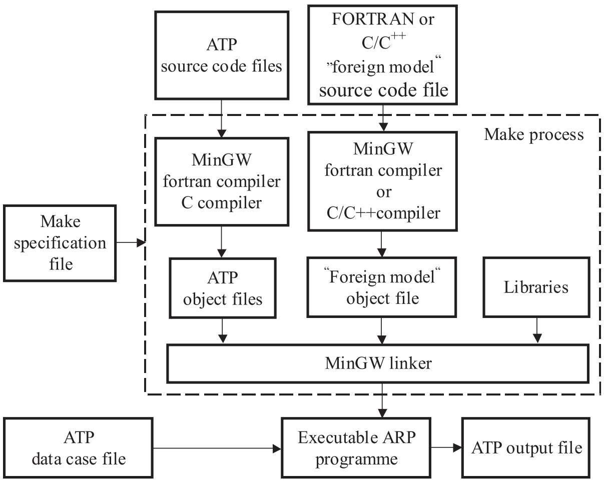

Fig. 1. Linking and compiling the process of ATP in MinGW32.

environment named, *eg*, *GCC=C:\gcc-2.95.2\bin* and *ATPMinG=C:\Eeug\LibMingw* with respective values have to be created. The whole process is clearly described in Fig. 1. Next, for faster execution it is suitable to create a batch file with content of *%gcc%make.exe<makefile*, whose parameters written in makefile file (Fig. 2) determine the parameters of this process.

```
CC=gcc
FOR=g77
OBJECTS =dimdef.o \
        newmods.o \
        comtac.o \
        userml.o \
        analyt.o \
        fgnmod.o \
        devt69.o \
        usrfun.o \
        hopcod.o \
        user10.o \
        cmodel.o \
        relay.o \ -name of file with digital relay model source
code

CFLAGS =  -DUNDERSCORE -O2
FFLAGS = -O2

IMAGE=tpbigm.exe

LIBRARY = tpbig.a C:\\dislin\\dismg7.a -luser32 -lgdi32 -
lcomdlg32

.f.o:
        $(FOR) -c $(FFLAGS) $<
.c.o:
        $(CC) -c $(CFLAGS) $(INCFLAGS) $<
$(IMAGE) : $(OBJECTS)
        $(FOR) -s -o $(IMAGE) $(OBJECTS) $(LIBRARY)
```

Fig. 2. Content of *makefile* file.

The file *makefile* makes a reference next on file *fgnmod.f* (Fig. 3), where MODELS language source code of "foreign model" can be externally placed or C++ language relay model functions saved in a stand-alone file have to be declared. The name of this file is defined in *makefile*.

Language MODELS is used for general description of the structure of models and the function of its elements. In comparison with high-level programming languages such as C++ its capabilities are considerably limited. To overcome these limits it includes "compiled foreign model" mechanism, which allows to connect an external model developed in another programming language and in this way eliminate its limitations. This mechanism can be used for development of a sophisticated digital relay model in a higher programming language and for connecting this model with ATP by a pre-defined interface "foreign model". This interface is defined by four data arrays storing the values of model data, inputs, outputs and variables. Every model must have initializing and execution procedures. Before calling procedures in MODELS language these must be linked and compiled with ATP. After this there can be created any number of instances of a given model working independently of each other in the simulation, with directives controlling simulation specified in the USE declaration.

```
SUBROUTINE FGNMOD ( name, namlen, xdata, xin, xout, xvar,
iniflg, ierflg)

    DATA refnam(3) / 'RELAY_C_MODEL' /

    ELSE IF ( iname.EQ.3 ) THEN
     IF (iniflg.EQ.1) THEN
       CALL c_relay_i(xdata, xin, xout, xvar)
     ELSE
       CALL c_relay_m(xdata, xin, xout, xvar)
     ENDIF

    RETURN
    END
```

Fig. 3. Example of "foreign model" definition in *fgnmod.f*.

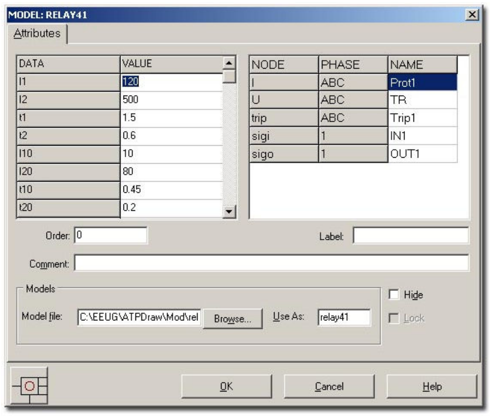

Fig. 4. Example of relay dialog window.

For a successful start of simulation with a newly compiled ATP we have to call from ATPDraw a batch file with this content *%atpming%tpbig.exe both%1%~n1. –R*.

3 MULTIFUNCTIONAL DIGITAL RELAY MODEL

Multifunctional digital relay model is created in three files *relay.sup*, *relay.mod* and *relay.c*. The first one represents component in ATP-Draw and creates an interface between the relay model and the user by which he can connect this relay into power system and input relay settings through the dialog window (Fig. 4). The second file

```
...
MODELS          - Beginning of Models section in the ATP data file
/MODELS
INPUT...        - definition of inputs from the power system network
OUTPUT...       - definition of outputs to the power system network
MODEL relay     - declaration of overcurrent relay model
TIMESTEP        - definition of sampling rate
INPUT           - definition of input instantaneous values of current for the relay model
DATA            - definition of relay characteristic settings
VAR             - definition of auxiliary variables
OUTPUT          - definition of relay model output
DELAY CELLS     - definition of delay cells size of instantaneous current past values
HISTORY         - definition of history values of current

MODEL C_RELAY FOREIGN RELAY_C_MODEL {ixdata:,ixin:,ixout:,ixvar:} - definition of
                foreign model of protection relay written in C++ file relay.c
EXEC
zfun(I2/I1):=(numerator/denominator) - expression of Z-transform transfer
function of low-pass discrete filte

USE C_RELAY AS C_RELAY   - use statement of relay foreign model
  DATA xdata[]:=[]       - set value of xdata[] for the foreign model
  INPUT xin[]:=[]        - load inputs xin[] for the foreign model
  HISTORY histdef(xvar[]):=[] - set history of xvar[] for the foreign model
  OUTPUT trip[]:=xout[]  - set output from xout[] for the foreign model
 ENDUSE
 ENDEXEC
 ENDMODEL

RECORD     - specification of variables to be plotted

USE relay AS relay
  INPUT ...              - definition of inputs for relay model
  DATA...                - definition of setting data for relay model
  OUTPUT...              - definition of outputs for relay model
 ENDUSE
ENDMODELS
```

Fig. 5. Structure of digital relay model in MODELS.

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <process.h>
#include <math.h>

int DFT(int m,double t,double dt, double *I, double T,int n, int md)
  { ... }

double RMS(int m1, double *X, int n)
  { ... }

int ZERO(int m1, double *X, double *Y, double *Z, double *A0, int
n)
  { ... }

int IMPEDANCE(int m1,double *I, double *U, double *Z)
  { ... }

int IMPEDANCE_ZERO(double *I0, double *U0, double *Z0)
  { ... }

int OVERCURRENT(double AR_t,double *ARtdelayI,double
timestep,double *trip,double I,double I1,double I2,double
*tdelayI,double t1,double t2)
  { ... }

int OVERCURRENT_I0(double AR_t,double *ARtdelayI0,double
timestep,double *trip,double I0,double fi0,double I10,double
I20,double *tdelayI0,double t10,double t20)
  { ... }

int DISTANCE_MHO(double AR_t,double *ARtdelayZ,double
timestep,double *trip,double *tdelayZ,double *Z,double
*MHOZ,double *MHOt, double Load_R,double Load_f,double
Line_a)
  { ... }

int DISTANCE_QUAD(double AR_t,double *ARtdelayZ0,double
timestep,double *trip,double *tdelayZ0,double *Z0,double
*ZA,double *ZB,double *ZC, double *QLR,double *QLX,double
*QLt,double Load_R,double Load_f,double Line_a,double Rfa)

 {
      DFT(m,time,timestep,SIA,T,NI,0);
      DFT(m,time,timestep,SIB,T,NI,(md.rem/3));
      DFT(m,time,timestep,SIC,T,NI,(2*md.rem/3));
      DFT(m,time,timestep,SUA,T,NU,0);
      DFT(m,time,timestep,SUB,T,NU,(md.rem/3));
      DFT(m,time,timestep,SUC,T,NU,(2*md.rem/3));
      DFT(m,time,timestep,SIBO,T,NI,0);
      DFT(m,time,timestep,SICO,T,NI,0);
      DFT(m,time,timestep,SUBO,T,NI,0);
      DFT(m,time,timestep,SUCO,T,NI,0);

      IA=RMS(m1,SIA,NI);  IB=RMS(m1,SIB,NI);
      IC=RMS(m1,SIC,NI);
      UA=RMS(m1,SUA,NU);
      UB=RMS(m1,SUB,NU);
      UC=RMS(m1,SUC,NU);

      ZERO(m1, SIA, SIBO, SICO, I0, NI);
      ZERO(m1, SUA, SUBO, SUCO, U0, NU);

      IMPEDANCE(m1, SIA, SUA, ZA);
      IMPEDANCE(m1, SIB, SUB, ZB);
      IMPEDANCE(m1, SIC, SUC, ZC);

      IMPEDANCE_ZERO(I0, U0 , Z0);

      DISTANCE_QUAD(AR_t,*ARtdelayZ0,timestep,*trip,*tdelayZ0,Z0,ZA,
      ZB,ZC,QLR,QLX,QLt,Load_R,Load_f,Line_a,RFa);
      DISTANCE_MHO(AR_t,*ARtdelayZ,timestep,*trip,*tdelayZ,ZA,MHOZ,
      MHOt,Load_R,Load_f,Line_a);
      DISTANCE_MHO(AR_t,*ARtdelayZ,timestep,*trip,*tdelayZ,ZB,MHOZ,
      MHOt,Load_R,Load_f,Line_a);
      DISTANCE_MHO(AR_t,*ARtdelayZ,timestep,*trip,*tdelayZ,ZC,MHOZ,
      MHOt,Load_R,Load_f,Line_a);

      OVERCURRENT_I0(AR_t,*ARtdelayI0,timestep,*trip,I0[0],Z0[1],I10,I
      0,*tdelayI0,t10,t20);

      OVERCURRENT(AR_t, *ARtdelayI, timestep, *trip, IA, I1, I2, *tdelayI,
      t1, t2);
      OVERCURRENT(AR_t, *ARtdelayI, timestep, *trip, IB, I1, I2, *tdelayI,
      t1, t2);
      OVERCURRENT(AR_t, *ARtdelayI, timestep, *trip, IC, I1, I2, *tdelayI,
      t1, t2);
 }

void c_relay3_i__(double xdata_ar[],
                  double xin_ar[],
                  double xout_ar[],
                  double xvar_ar[])
{ ... }
```

Fig. 6. Structure of digital relay model in C++.

represents part of relay model created in MODELS language. A sampling circuit and a low-pass digital filter are created in it. The structure of the relay model in MODELS language is in Fig. 5.

Next it submits all data to relay model created in C++ language by four data arrays, located in the third file. The structure of relay model C++ source code is in Fig. 6.

Interconnection of the relay model with the power system is made in section Models. This interconnection is represented by input values of the branch currents and voltages in individual phases measured through a circuit breaker and by the output value signals for disconnecting the circuit breaker and overreaching signal for the second relay at the other end of the line. The exchange of values between individual sections creates a hierarchical structure. The names of respective nodes in the power system are defined in INPUT and OUTPUT directives of Models section. The next values of Models section are available in the relay model by directives INPUT and OUTPUT in respective variables defined in Mode relay section. The re-

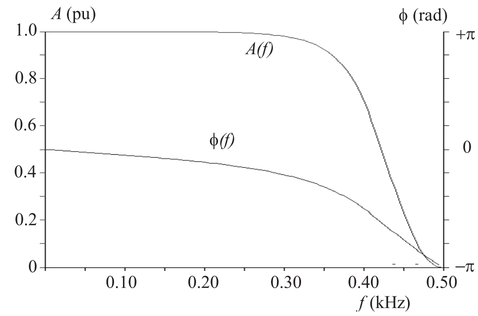

Fig. 7. Frequency behaviour of the Butterworth second order filter, amplitude and phase.

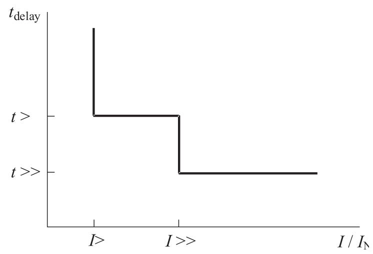

Fig. 8. Double stage definite-time overcurrent characteristic

quired values by "foreign model" are available from Model relay section by INPUT and OUTPUT directives defined in USE statement and given by four arrays *xdata*, *xin*, *xout* and *xvar*.

3.1 Sampling circuit

The sampling circuit is created by TIMESTEP directive, which determines the instants when the model is called during the simulation. In this way, values are gained in regular time intervals from the network and a required number of samples are stored in the buffer created by DELAY CELLS directive, depending on the size of the time-step. This samples are needed for filter operation and Fourier analysis of each analyzed quantity.

3.2 DIGITAL FILTER

As a digital filter in the relay low-pass Butterworth second order filter was chosen [9, 10]. It is represented by a transfer function in z-area acquired from the transfer function in Laplace area by z-transform [2]. Respective function in the model is entered by procedure ZFUN in the form of: ZFUN(Y/X)=NUMERATOR/DENOMINATOR.

Transfer function in z-transform for Butterworths filter gererally has form

$$H(z) = \frac{\sum_{k=0}^{n} a_k z^{-k}}{1 + \sum_{k=1}^{n} b_k z^{-k}} \tag{1}$$

and for second order filter

$$a_0(k) = a_2(k) = \frac{\Omega_c^2}{c(k)}\,, \quad a_1(k) = 2a_0(k)\,,$$

$$b_1(k) = \frac{2\big(\Omega_c^2 - 1\big)}{c(k)}\,,$$

$$b_2(k) = \frac{1 - 2\cos\big(2k+1)\pi/(2N)\big)\Omega_c + \Omega_c^2}{c(k)}\,,$$

$$c(k) = 1 + 2\cos\big(2k+1)\pi/(2N)\big)\Omega_c + \Omega_c^2\,,$$

$$\Omega_c = \tan\Big(\frac{\pi}{f_r}\Big), \quad f_r = \frac{f_s}{f_c}\,, \quad N - \text{order of the filter},$$

$$k - \text{elementary filter number } k = 0 - \frac{N}{2} - 1\,,$$

$$f_c - \text{cut off frequency}, \quad f_s - \text{sampling frequency}.$$

Frequency dependent behaviours of the amplitude and phase for a second order Butterworth filter are shown in Fig. 7 for cut-off frequency 200 Hz and sampling frequency 500 Hz.

3.3 Processing of filtered input signals

Input signals filtered from higher harmonics are next passed to relay model in C++, where they are processed by discrete Fourier analysis [1], [3]. Filtering is necessary for increasing the accuracy of calculating the Fourier coefficients. A discrete Fourier analysis algorithm processes the samples of input behaviours sampled in time intervals equal to value of TIMESTEP directive. It uses floating window, which means that calculation of Fourier series coefficients is executed after every new sample and is not waiting for whole period to be sampled.

$$a_n = \frac{2}{T}\sum_{n=0}^{m} f(t)\Delta t \cos\frac{2\pi nt}{T} \tag{2}$$

$$b_n = \frac{2}{T}\sum_{n=0}^{m} f(t)\Delta t \sin\frac{2\pi nt}{T} \tag{3}$$

where a_0, a_n, b_n — are coefficients of the Fourier series, $f(t)$ — is functional value of signal sample in time t, Δt — is value of sampling time step, T — is signal period, m — the number of samples in one period, n — harmonic number, in the case of even number of samples $n = m/2$ while for m odd $n = (m-1)/2$ and m - is the number of samples in one period.

The final behaviour is formed by,

$$f(t) = \frac{1}{2}a_0 + \sum_{n=1}^{k} a_n \cos\frac{2\pi nt}{T} + \sum_{n=1}^{k} b_n \sin\frac{2\pi nt}{T} \tag{5}$$

or,

$$f(t) = \frac{1}{2}a_0 + \sum_{n=1}^{k} \sqrt{a_n^2 + b_n^2}\,\sin\Big(\frac{2\pi nt}{T} + \arctan\frac{b_n}{a_n}\Big) \tag{6}$$

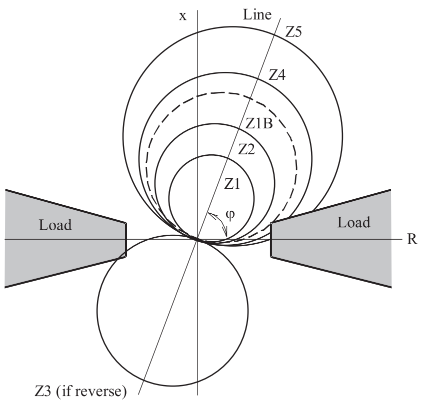

Fig. 9. Offset MHO distance relay tripping characteristic.

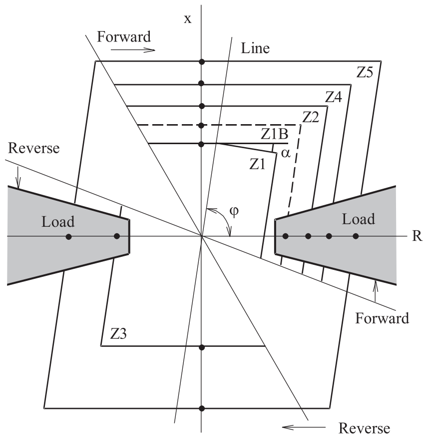

Fig. 10. Quarilateral tripping characteristic of ground distance relay.

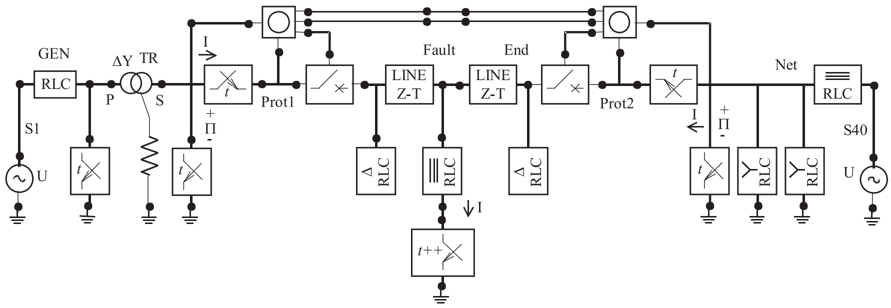

Fig. 11. Example of power system and relay connection.

from which by definition of the root mean square value we get

$$I_{RMS} = \sqrt{\frac{a_0^2}{4} + \sum_{n=1}^{k} \frac{a_n^2 + b_n^2}{2}} \tag{7}$$

The way of operation of the multifunctional digital relay model in C++ is that in its every execution it calls functions for Fourier series coefficients calculation and RMS values of currents and voltages stated in the main part. The impedance in each phase, current and voltage zero symmetrical components are calculated from them by calling respective functions. These values are input parameters for relay protection functions where they are evaluated.

The multifunctional relay includes a non-directional overcurrent relay with double set-stage tripping characteristic as shown in Fig. 8. It is suitable to set it as a backup relay for other relays [11, 12]. It monitores values of currents in each phase and in the case of exceeding the set limit it activates the tripping signal after a respective elapsed time. In the case of one-phase automatic reclosure setting after an entered time it activates the signal for switch closing, and if the fault is permanent after an instantaneous pre-set time stage (for the reason of nonsensitivity to emerging transient phenomena) the definite switch-off signal is generated. In the case of fault clearing during auto-reclosure time interval the auto-reclosure automatic is blocked for next 5 seconds. Each relay can be used individually or any combination from them depends only on its settings.

The second relay is a directional overcurrent relay with a double set-stage tripping characteristic [13, 14] shown in Fig. 8 for zero sequence symmetrical current component. It is suitable for ground fault protecting during which for a zero sequence current emerges because of nonsymmetry. It also includes a three phase auto-reclosure automatic.

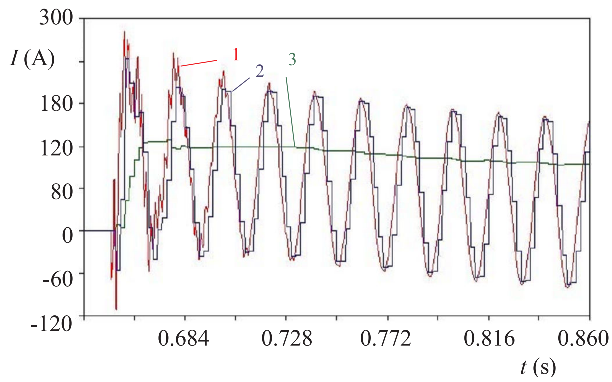

Fig. 12. Time behaviour of instantaneous current value and the output from digital filte: 1 — short circuit current in phase A, 2 — the same at the digital filter output, 3 — RMS value calculated by relay.

The next one is a distance relay with a tripping characteristic offset MHO as shown in Fig. 9. In order to guarantee reliable discrimination between load operation and

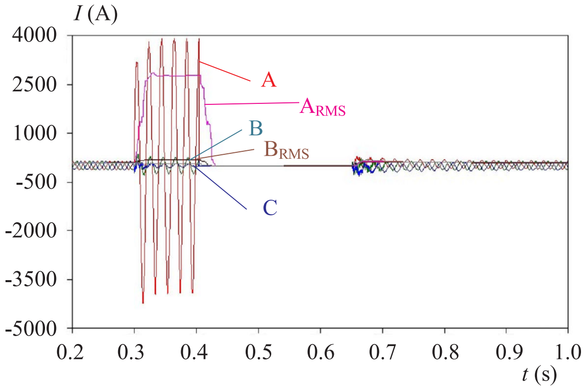

Fig. 13. Time behaviour of instantaneous and RMS values of currents at the beginning of line in phases A,B,C, after one-phase short.

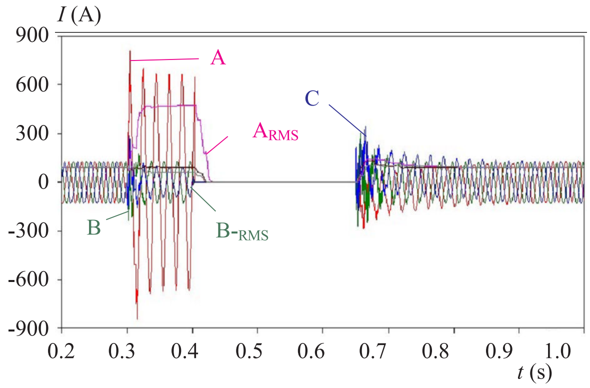

Fig. 14. Time behaviour of instantaneous and RMS values of currents at the end of line in phases A,B,C, after one-phase short.

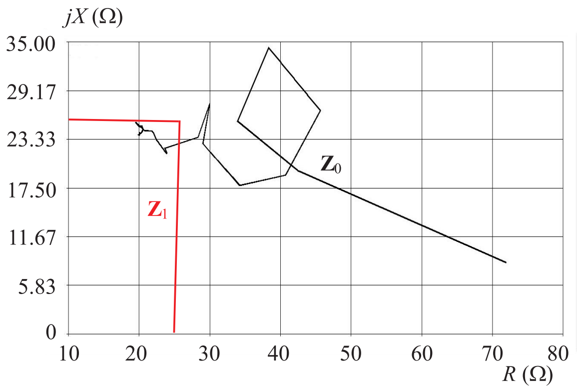

Fig. 15. Behaviour of zero impedance symmetrical component seen by relay at the beginning of line; $\boldsymbol{Z}_0$ — zero impedance locus calculated by relay, $\boldsymbol{Z}_1$ — border of the 1st distance zone.

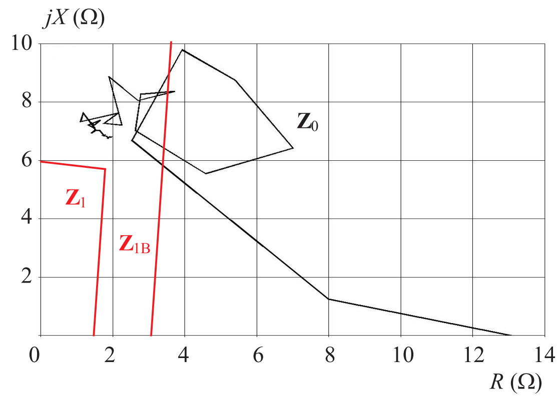

Fig. 16. Behaviour of zero impedance symmetrical component seen by relay at the end of line; $\boldsymbol{Z}_0$ — zero impedance locus calculated by relay, $\boldsymbol{Z}_1$ — border of the 1st distance zone, $\boldsymbol{Z}_{1\mathrm{B}}$ — border of the overreached distance zone.

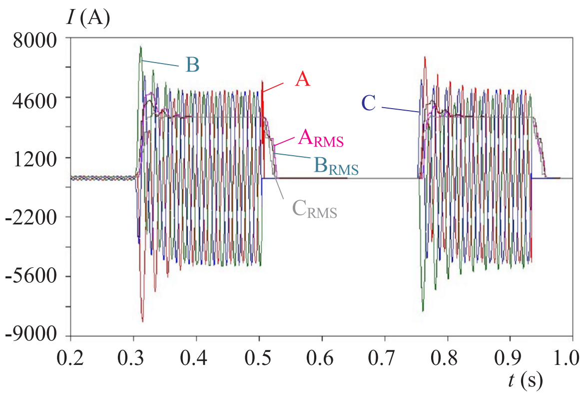

Fig. 17. Time behaviour of instantaneous and RMS values of currents at the beginning of line, after three-phase short.

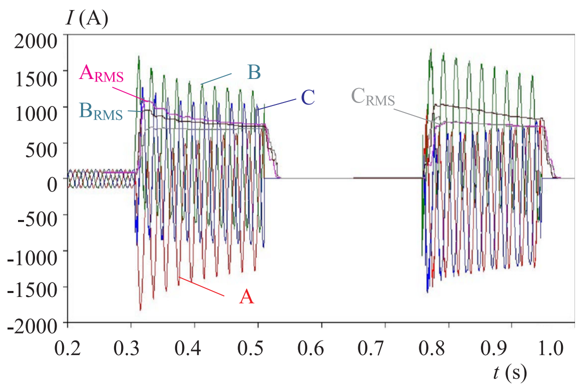

Fig. 18. Time behaviour of instantaneous and RMS values of currents at the end of line, after three-phase short.

short-circuit — especially on long high loaded lines — the relay is equipped with a selectable load encroachment characteristic. Impedances within this load encroachment characteristic prevent the distance zones from unwanted tripping. This relay is next equipped with one phase auto-reclosure automatic and in the case of fault in the first distance zone it activates a signal for the opposite relay at the other end of the line to overreach the first distance zone, which ensures a switching-off the fault on the line in instantaneous time and enables auto-reclosure automatic for successful fault clearing.

The last relay is a ground distance relay with a quadrilateral tripping characteristic [14] shown on Fig. 10. It is used for clearance of ground faults; it includes three phase auto-reclosure automatic, load areas and overreaching distance measuring function.

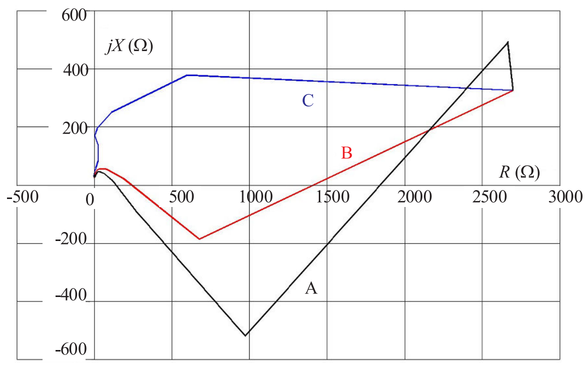

Fig. 19. Behaviour of line impedances seen by relay at the beginning of the line. A, B, C - impedance locus calculated for individual phases

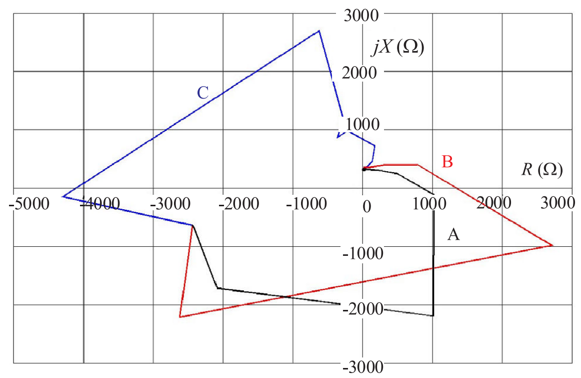

Fig. 20. Behaviour of line impedances seen by relay at the end of the line. A, B, C - impedance locus calculated for individual phases.

4 APPLICATION

As an example of a correct function of the relay model, a simulation of a power system and relay connection ashown in Fig. 11 is presented next. This example is modelled in ATPDraw with ground faulted phase A in 10 % distance of line length from line beginning, arising at time 0.3 s. The function of the low-pass digital filter is shown in Fig. 12 as time bahaviour of current and its RMS value calculated by relay. Time bahaviours of currents and current RMS values calculated by relays at the beginning and at the end of line are in figures 13 and 14.

From the above behaviours it is evident that the relay at the end of the line tripped in overreaching distance zone, activated by the opposite relay, in instantaneous time. Three phase auto-reclosure automatic timer was activated with time interval 0.25 s, which after successful fault clearance connected the line to power system and after decay of the transient phenomena the system continued in stable operation. After this, the auto-reclosure automatic was blocked for 5 seconds. From the behaviours of the zero sequence impedance in figures 15 and 16 it is clear that the tripping signal was activated by ground distance relay with setting the first distance zone as 85% of the line length and the time delay of 0.1 s on the first relay to $R1 = 23.8\,\Omega$, $X1 = 24.65\,\Omega$ and on the second relay to $R1 = 1.7\,\Omega$, $X1 = 5.95\,\Omega$.

In the second case a three phase fault on line was simulated in 10 % distance of line length from the line beginning. Time behaviours of currents and current RMS values calculated by relays at the beginning and at the end of line are in figuresn 17 and 18.

We can see from behaviours in Figs. 17 and 18 that both relays tripped in instantaneous time set-stage 0.2 s and the second relay overreached the first zone after receiving a signal from the first one. After switching-off the line, the auto-reclosure automatic timer was activated and after 0.25 s it connected a permanent faulty line, which resulted in instantaneous definite line disconnection. From the behaviours of each phase impedance on Figs. 19 and 20 it is evident, that in the multifunctional relay the distance relay tripped this fault with setting of the first distance zone to 85% of the line length and with teh time delay stage 0.2 s in the first and the second relay to $Z1 = 295\,\Omega$.

5 CONCLUSION

This paper described a powerful approach to simulation of an interactive protection system, which is an ATP simulation program connected with a "foreign model" programmed in high programming language C++ appropriate for development and testing of new relay algorithms. Even though general information about linking of the "foreign model" programmed in C++ language and compiling with ATP is in *readme.txt* files on EEUG web pages, the first-time user can spend a long time with setting everything under proper control. Therefore this article gives specific information about linking of C++ model and compiling with ATP. The developed multifunctional digital relay is presented here with described individual components. Finally, correct performance of the multifunctional digital relay is verified by simulation. The future work will be the development of this digital relay and new protection algorithms for protecting medium voltage networks with an isolated or resonant earthed neutral point.

Acknowledgement

This project was supported by the Slovak Grant Agency under Grant No. 1/3092/06.

References

[1] ANDERSON, P.M.: Power System Protection, IEEE Press, New York, 1995.

[2] JANÍČEK, F.—CHLADNÝ, V.—BELÁŇ, A.—ELESCHOVÁ, Ž.: Digitálne ochrany v elektrizačnej sústave, Vydavateľstvo STU, Bratislava, 2004. (in Slovak)

[3] MOORE, P.J.: Power System — Protection, The Institution of Electrical Engineers, London, 1995.

[4] DOMMEL, H. W.: Electromagnetic Transients Program (EMTP) Theory Book, Bonneville Power Administration, Portland, OR, 1986. Available on internet: http://www.eeug.org/files/secret/EMTP_TheoryBook .

[5] CanAm EMTP User Group, Alternative Transient Program (ATP) Rule Book, Portland, OR, 2001. Available on internet: http://www.eeug.org/files/secret/ATP_RuleBook .

[6] DUBÉ, L.: MODELS in ATP, Language Manual, 1996. Available on internet: http://www.eeug.org/files/secret/MODELS .

[7] SINTEF Energy Research, ATPDRAW Version 3.5 Users' Manual, Trondheim, Norway, 2002. Available on internet: http://www.eeug.org/files/secret/ATPDraw .

[8] MinGW User Group, Introduction to MinGW, Available on internet: http://www.mingw.org .

[9] ONDRÁČEK, O.: Signály a sústavy, Vydavateľstvo STU, Bratislava, 2003. (in Slovak)

[10] ONDRÁČEK, O.: Diskrétne signály a sústavy, Vydavateľstvo STU, Bratislava, 2000. (in Slovak)

[11] ELESCHOVÁ, Ž: The Digital Protection Relays, Proceeding of the Second Conference on Electrical Enginnering and Information Technology for PhD students ELITECH '99. FEI STU Bratislava, 1999.

[12] SMOLA, M.—BELÁŇ, A.—ELESCHOVÁ, Ž.—JANÍČEK, F.: Power Lines Protection by Digital Relays, Proceeding of Abstract of the 6th International Conferences "Control of Power Systems '04". Štrbské Pleso, 2004.

[13] CABALA, M.—BOHÁČ, M.: Assessing Stability Condition to Evaluate the Risk of Blackout. (Hodnotenie podmienok a okolnosti stability sústav pre stanovenie miery rizika výskytu BLACKOUT-u), In: Elektroenergetika 2005. TU Košice, 2005, CD-Rom, 3rd International Scientific Symposium "Elektroenergetika 2005", Stará Lesná. Slovak Republic, 21–23 September 2005.

[14] FECKO, Š.—BOHÁČ, M.: Compact Power Lines, In: Elektroenergetika 2005. TU Košice, 2005, CD-Rom, 3rd International Scientific Symposium "Elektroenergetika 2005", Stará Lesná. Slovak Republic, 21–23 September 2005.

[15] TOMAN, P.—HALUZÍK, E.: Location of Earth Faults in a High Voltage Networks, In: Sborník prací studentů a doktorandů, Akademické nakladatelství CERM, s.r.o., 2000, pp. 118–120.

[16] TOMAN, P.—HALUZÍK, E.: Location of Earth Faults in Compensated MV Distribution Networks, In: 1. Mezinárodné vedecké sympózium Elektroenergetika 2001. Košice: Fakulta elektrotechniky a informatiky, TU Košice, Katedra elektroenergetiky, pp. 259–261.

[17] TOMAN, P.—ORSÁGOVÁ, J.: Electrical Power System Simulation, In: The 2nd International Scientific Symposium ELEKTROENERGETIKA 2003. Košice: Mercury – Smékal Publishing House, 2003, pp. 196–203.

[18] RAPŠÍK, M.—SMOLA, M.—BOHÁČ, M.—MUCHA, M.: Basics of Power Engineering I, STU Bratislava, FEI, 2004.

[19] BELÁŇ, A.—ELESCHOVÁ, Ž.—SMOLA, M.: Rezonančné prepätia v elektrických sieäch, EE — Časopis pre elektrotechniku a energetiku **X**, 35–38, mimoriadne číslo, 2004.

[20] SMOLA, M.—BELÁŇ, A.—ELESCHOVÁ, Ž.: Power Lines Protections by Digital Relays, Zeszyty Naukowe Nr 299/2005 — Elektryka z. 55. Wydzial Elektrotechniki i Automatyki, Politechnika Opolska 2005. pp. 203–208.

[21] BELÁŇ, A.—ELESCHOVÁ, Ž.—SMOLA, M.: Resonance Overvoltages in Electric Power Networks, Zeszyty Naukowe Nr 299/2005 — Elektryka z. 55. Wydzial Elektrotechniki i Automatyki, Politechnika Opolska 2005. pp. 33–40.

[22] BELÁŇ, A.—ELESCHOVÁ, Ž.: Selected Problems of Turbogenerators Protections, Zeszyty Naukowe Nr 299/2005 — Elektryka z. 55. Wydzial Elektrotechniki i Automatyki, Politechnika Opolska 2005. pp. 41–56.

[23] BELÁŇ, A.—ELESCHOVÁ, Ž.: The False Operation of the Tank Earth-Fault Protection of the Block Transformer, In: Proceedings of the 5th WSEAS Conference POWER'05. 5th WSEAS/IASME International Conference on: ELECTRIC POWER SYSTEMS, HIGH VOLTAGES, ELECTRIC MACHINES (POWER'05). Tenerife, Spain: WSEAS,.

Received 30 March 2006

František Janíček (Prof, Ing, PhD) graduated in power engineering from the Slovak University of Technology in 1979. In 1984 he gained the PhD degree. In 1999 he was appointed Professor in power engineering. Since 1990 until now he has been a member of the Scientific Council FEI, and since 1995 a member of the Supervisory Board of the Western Slovak Energy Distribution Company. He is the founding member of the Slovak Committee of the World Energy Council. Professor Janíček is the Head Manager of the Project "Qualification of Selected Electric Equipment of the Nuclear Power Plants in Slovakia and in the Czech Republic". Since 2000, Professor Janíček has been the Dean of the Faculty of Electrical Engineering and Information Technology of the Slovak University of Technology in Bratislava.

Martin Mucha (Ing) received his BE and ME (Ing) degrees with honor from Slovak University of Technology, both in power engineering in 2002 and 2004 respectively. He is with Slovak University of technology pursuing his PhD degree since October 2004. His research interests are power system protection techniques, substation automation, modelling and simulation methods in power systems, overvoltage study and overvoltage protection.