Automated Generation of Fault Case Sets for the Functional Test of Numerical Distance Protection Relays

R. Huwer
huwer@e-technik.uni-kl.de
D. Nelles
dnelles@e-technik.uni-kl.de
Department of Power Systems
University of Kaiserslautern
67653 Kaiserslautern, Germany

M. Igel
100565.1206@compuserve.com
AEG ATLAS
Protection and Control
60528 Frankfurt, Germany

Abstract — **Testing of protection relays is an important element of quality assurance. In this article a computer aided functional test to support the functional test of numerical distance protection relays is presented. The complete test fault case sets are generated automatically using a flow diagram as a behaviour model of the relay. The number of test fault cases is reduced by using additional information about the relay's inner structure. The test cases are simulated in a power network of minimal size using ATP.**

Keywords: relay test, functional test, equivalence class test, behaviour model, flow diagram, ATP

I. INTRODUCTION

Distance protection relays scan the local currents and voltages and calculate an impedance which is proportional to the fault distance. This impedance is compared with a tripping characteristic. If the fault impedance is less than the tripping impedance the power-off command is given to the circuit breaker.

The root of a numerical protection relay is basically a mathematical processor which runs a program that is stored in a ROM. Its process interfaces consist of analog input and output channels as well as serial interfaces. Because the failure of protection relays leads not only to blackouts of the power supply but can also endanger human life, the protection relay has to pass numerous tests to prove its correct behaviour. One part of protection testing is the test of each individual functionality and the intermodal behaviour of all components. The usage of an actual test strategy is the base of protection testing. A goal is to increase the satisfaction of the client and to decrease the number of software updates [10].

Since the probability of fault presence in a software product increases with the program's size and complexity [6], the number of test cases to check the software's correctness increases. Manufacturers use different techniques and tools to inspect their products and to minimize the probability of software bugs. One possibility to test a protection device is to generate test fault cases, stimulate the protection relay with them and analyze the relay's behaviour.

This article describes a technique which generates the complete set of test cases which are needed to check the protection relay automatically. The difficulty is to obtain the complete test case set which includes the minimum number of necessary test fault cases. Because of the large number of test cases it is necessary to reduce the test case sets using additional information about the relay's interior structure. The shapes of the line-to-earth voltages and line currents at the relay's location are then simulated for each test fault case in a test network using the electromagnetic transient program ATP. Metrics allow the estimation of the progression of the test process.

II. TESTING TECHNIQUES

The analysis and classification of software errors is significant for the development of error-free programs. The types of software errors may be devided into functional errors, system errors and runtime errors [2]. Because of functional errors a program executes incorrect operation or no operation. System errors lead to a false behaviour of the systems hardware. Runtime errors may be observed for example when a division by zero occurs which is not inhibited.

Each testing technique possesses of a qualification to detect special types of errors. The known techniques may be devided into dynamic testing techniques, static analysis, symbolic testing techniques and verification [5].

During the test the source code of the software is not viewed. Therefore the protection relay acts like a 'black box' and the only usable testing technique is a dynamic test. Dynamic techniques are a common type of system testing. However they require the test with concrete test cases. The goal is to obtain a comprehensive test with a minimum number of test cases. Dynamic testing techniques all have random sample character, hence they can not prove that the test object is bug-free. Apart from this disadvantage the dynamic operation examines the hardware interfaces and the interaction of all device components during the test.

The functional test checks the protection relay in relation to the guide book of the manufacturer. Thereby it may happen that some lines of the program source code are not executed. Because of this fact, the functional test cannot replace further tests where each program line is executed at least one time.

III. STRUCTURE OF A NUMERICAL DISTANCE RELAY

The software which is implemented in a digital protection relay [1] may be described by the combination of several processes (PRO). A simplified scheme is shown in Figure 1. After the currents and voltages of the power network have been sampled by the input transformer and have been converted to numerical variables they appear at the input signal of the PRO 'starting system' A. If this process detects a violation of the adjusted limits of currents or voltages the starting signals are sent up to the PRO 'measurement loop selection' B and the PRO 'timer' C is started. Because of the starting signals, process B selects one current and one voltage. With these signals the PRO 'impedance algorithm' D calculates an impedance $\underline{Z}$ which is proportional to the fault distance. The result is transmitted to the PRO 'tripping characteristic' E and the PRO 'fault direction' F. If the time C elapses, the calculated impedance is in the tripping characteristic line and the fault direction accords to the adjusted direction, the PRO 'tripping command' G gives the order to switch-off the circuit breaker.

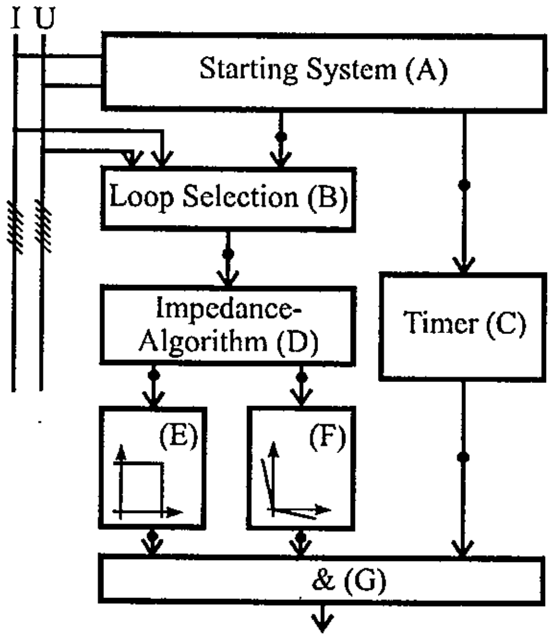

Fig. 1: Topology of the processes

IV. BEHAVIOUR MODEL OF THE RELAY

To check the functional correctness of the protection relay it has to pass a large number of test cases. These test cases are generated automatically with the assistance of a computer. Therefore the desired behaviour of the protection relay is modelled as a flow diagram in a computer readable manner. Each process is simulated as a behaviour model, which may have several inputs and outputs. Because of the utilization of unique names of the processes and its inputs and outputs, the process structure may be handled by the computer. The several processes may be combined in any way. To achieve this the output of a process and the input of the next process are combined at a node. Each node may have several attributes. The most important are:

- node is observable from outside the protection relay
- node is not observable from outside the protection relay
- node signal belongs to phase L1
- node signal belongs to phase L2
- node signal belongs to phase L3

After having captured the process topology the tester specifies which node signal has to be checked. This node must be observable from outside the relay. A search algorithm traces the signals backwards until all input nodes are observable. By this way the hull obtained covers a minimum number of processes.

V. AUTOMATED GENERATION OF TEST CASES

The method of generating test cases is based on an equivalence class test. The range of the program values is divided into equivalent classes, such that the classes of value states build disjunct sets. If the program module for example includes the request

if (y > 2) then < statement >

the range of the value y is devided into two disjunct sets with the value states $y > 2$ and $y \leq 2$. The test cases have to be generated for both states. Because the request results in the value TRUE if the test value is $y = 3$ or $y = 4$, the correct functionality is demonstrated using only one test case within one class.

The behaviour of a process, which is normally more complicated, is defined by giving the value of the output variable for each combination of the input variables. If the tester wants to emulate the logical AND-function (Figure 2) he has to specify the values of the output variable A for all four combinations of input value classes.

Name: AND
Input: E_1, E_2
Output: A

Fctn:	FALSE	FALSE	FALSE
	TRUE	FALSE	FALSE
	FALSE	TRUE	FALSE
	TRUE	TRUE	TRUE

The number of test cases TM of a complete test of a process PRO results by multiplication of the number of equivalence classes K_i of all input values E_i:

$$TM = \prod_i K_i \tag{1}$$

In this example the number of test cases of the And-function is computed by:

$$TM = \prod_{i=1}^{2} K_i = K_1 \cdot K_2 = 2 \cdot 2 = 4 \tag{2}$$

If j different processes are combined, the total number of theoretical possible test cases TS is calculated by:

$$TS = \prod_j TM_j \tag{3}$$

The term 'complete test' means that each additional test case which is not included in the test case set gives no additional information about the protection relay's behaviour.

The described AND-function possesses no time-dependent behaviour. If processes which include signal delays or memory elements are tested, the behaviour of the module depends on its history. Hence it is necessary to prove the correct behaviour in the time domain. Therefore the protection relay is stimulated with the combination of two test cases out of the test case set. The number of these fault combinations TD is given by:

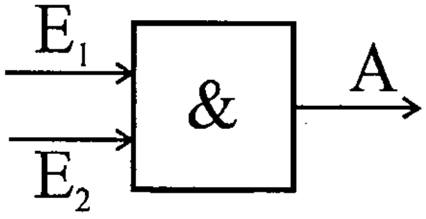

Fig. 2: AND-Function

$$TD = TS \cdot (TS - 1) \approx TS^2 \tag{4}$$

Inside the resulting test case set, the commonly used fault cases like evolving faults or cross-country faults are already included. The combination of test cases is limited to two cases. If three test cases would be combined, the number of test case combinations would be obtained by TD ≈ TS^3.

VI. REDUCTION OF THE NUMBER OF TEST CASES

By combination of several processes within a minimal hull a large number of test cases is obtained. Therefore it is reasonable to reduce the number of test cases within the test case set using additional conditions (informations). One such condition is the fact that a certain combination of variable states cannot appear at all. All test cases which include these state combination may be deleted out of the test case set.

A further condition is the usage of typical attributes of the protection relay. Assuming that the PRO 'measurement loop selection' has a cyclic symmetry in regard to the three phases L1, L2 and L3 of the power network. The relay's behaviour is indifferent to whether a short-circuit occurs in phase L1 or in phase L2. Consequently it is possible to reduce the number of single test cases to a third. The largest number of test case reductions is obtained if the number of observable nodes is increased because the size of the minimal hull is decreased. The several techniques which the reduction algorithm uses are stored in a database as the knowledge of an expert. The database may be modified or upgraded at any time.

To decrease the test duration it may be required to test only a subset of all test cases. Using the ratio of the number of actual executed test cases TS_{act} (resp. TD_{act}) to the number of all test cases TS (resp. TD), the metrics of the complete test case coverage CTS (resp. CTD) specify the proportion of all test cases that are performed. This metrics are defined as:

$$\begin{aligned} CTS &= TS_{act}/TS \\ CTD &= TD_{act}/TD \end{aligned} \tag{5}$$

They may be calculated for single processes as well as for multiple processes. It is even possible to estimate the time which is needed to check the remaining test case set.

VII. SIMULATION OF TEST FAULT CASES

The set of generated test cases has to be simulated one by one as short circuit in a test network. This test network has to fulfill some basic requirements:

- All test cases within the test set shall be simulated in the same test network in order to obtain comparable test results.
- The test network shall have a minimal size.
- Non-ideal attributes of the network are neglected if they don't trigger special functions in the protection device.

The test network in Figure 3 fulfills the previous conditions and can be used to test the numerical protection relay. It consists of a three-phase network where two power supplies are connected by two lines. Thus it is possible to simulate an electrical power transmission during the duration prior to a fault. The numerical distance protection relay is located at the black dot P1 on line 2. By this way the simulation of the short circuit is possible in the forward- and the backward-direction of the relay.

At the location of the distance protection relay the line currents and the line-to-earth voltages are simulated using ATP. To create the ATP-input file the tester may utilize a graphical user interface. The network that has to be simulated in the next test case may be adjusted directly on the screen by mouse- and/or keyboard-control. Next the ATP-input file is generated automatically in ASCII-format. To allow an automated cycle of the test series it is possible to control the graphical user interface in batch mode using a fault description language.

The test of a numerical protection relay may be divided into several parts. The root of the protection relay's software is the distance protection algorithm. To test this algorithm the relay is stimulated with test cases which include non-ideal effects like harmonics, current transformer saturation, etc [4,7]. As result the 'uncertainty of the protection relay' is obtained [8,9]. Non-ideal effects in the test network like unsymetrical lines, mutual impedances, counter infeeds or fault resistances lead to the 'uncertainty of the network' and have to be considered by the relay's settings [11]. Dynamic faults like evolving faults or cross-country faults may lead to an undesired behaviour of the protection relay, wherefore it has to be tested with these test fault cases.

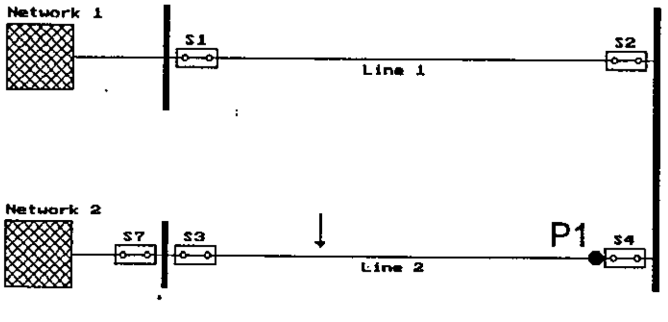

Fig. 3: Minimal test network

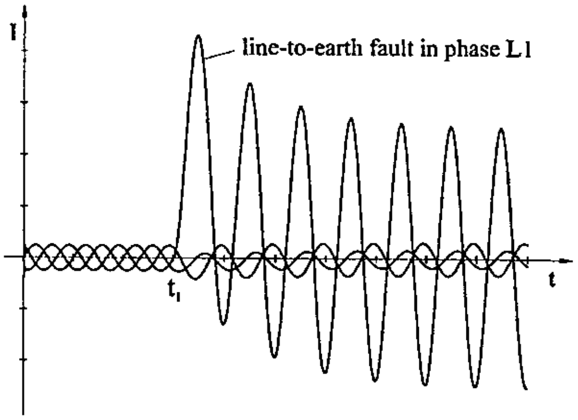

Fig. 4: Shape of line currents at the relay's location

Because only the protection relay's functionality is checked, the test is executed using ideal input signals. Hence the relay's input signals are not impressed with non-ideal effects like harmonics or effects of current transformer saturation etc. as usual. If the functional test would be performed using non ideal input signals for example faults near the tripping border, it would be impossible to locate the cause of defect in the case of a faulty behaviour. Consequently non ideal input signals would falsify the relay's reaction.

Each test fault case is simulated in the given network. Figure 4 shows the shapes of the line currents of an one-pole line-to-earth fault in the middle of line 2 as seen by the relay. The test fault case is divided into a prefault time ($t \prec t_1$) and the fault time ($t \succeq t_1$). During the prefault time the relay may obtain a steady state. While the fault time the relay's behaviour is recorded and it's actual behaviour is compared with the desired behaviour.

IIX. EXAMPLE

The presented method is now demonstrated with an easy example. A simplified part of a digital protection relay is examined (Figure 5). Only the connections of the processes 'measurement loop selection' B, 'impedance-algorithm' D and 'tripping characteristic' E are taken into account. It is

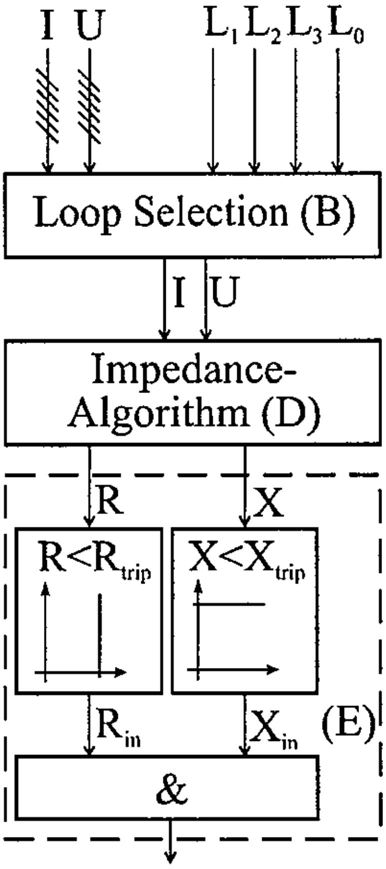

Fig. 5: Simplified distance protection relay

supposed that the input signals of the PRO 'measurement loop selection' B which have already been checked in previous test series have a time-dependent behaviour. The impedance-algorithm has also been checked in previous tests, its attributes are known [8,9]. The result of the algorithm, the calculated impedances R and X are admitted to the PRO 'tripping characteristic' E which has a time-independent behaviour. A process which recognizes the fault direction is not available.

Because only the input variables of the PRO 'measurement loop selection' and the output signals of the PRO 'tripping characteristic' are observable from outside the relay, the combination of processes represents the minimal hull. The input signals of the PRO 'measurement loop selection' are the starting signals L1, L2, L3 and L0. They are boolean variables which adopt the value TRUE or FALSE depending on the type of short circuit. For each combination of these variables it is determinated in a decision table (Table 1) which one of the three line currents and line-to-earth voltages is transmitted to the algorithm. After calculating the fault impedance by the algorithm the impedance $\underline{Z}$ is devided into its real component R and its imaginary component X. The PRO 'tripping characteristic' examines if both components comply the conditions $R < R_{trip}$ and $X < X_{trip}$. If both values are less than the tripping impedance the variable of the output signal obtains the value TRUE. The behaviour of the whole system is shown in Figure 5. The disjunct classes of the several program variables are defined in Table 2.

Tab. 1: Behaviour of the PRO 'measurement loop selection'

L1	L2	L3	L0	I	U
1	0	0	1	I1	U1
0	1	0	1	I2	U2
0	0	1	1	I3	U3
1	1	0	0	I1-I2	U1-U2
0	1	1	0	I2-I3	U2-U3
1	0	1	0	I3-I1	U3-U1
1	1	0	1	I1-I2	U1-U2
0	1	1	1	I2-I3	U2-U3
1	0	1	1	I3-I1	U3-U1
1	1	1	0	I1	U1
1	1	1	1	I1	U1

Tab. 2: Disposition of equivalence classes

name	classes		TM
L1	TRUE	FALSE	2
L2	TRUE	FALSE	2
L3	TRUE	FALSE	2
L0	TRUE	FALSE	2
R	< Rtrip	> Rtrip	2
X	< Xtrip	> Xtrip	2
Rin	TRUE	FALSE	2
Xin	TRUE	FALSE	2

Using (3) the maximum number of theoretical test cases is obtained by:

$$TS = \prod_{j=1}^{8} TM_j = 2^8 = 256 \qquad (6)$$

As a result of the time dependence of the PRO 'measurement loop selection' the whole system becomes time-dependent, too. The maximum number of theoretical fault combinations is calculated by (4):

$$TD = TS \cdot (TS - 1) = 256 \cdot 255 = 65280 \qquad (7)$$

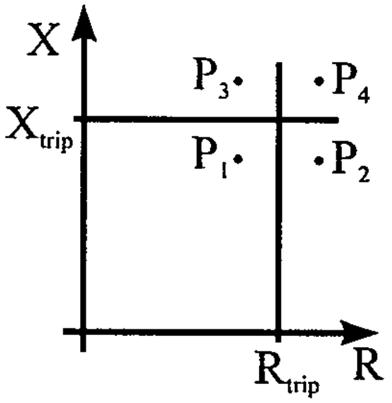

Fig. 6: Test points of the tripping characteristic

Although the example is very simple the number of test cases is already very large. Therefore the usage of reduction techniques is required.

Usage of reduction techniques

Until now only the maximum test case set that is obtained combining all classes of variables was mentioned. However the real set is smaller. For example the PRO 'measurement loop selection' does not have 16 input combinations of the variables L1, L2, L3 und L0, but only 11 cases (Table 1). Other combinations cannot occur. Looking at the PRO 'tripping-characteristic' its recognizable that not all combinations of variable classes are possible either. The number of necessary combinations is reduced from 16 to 4 corresponding to the test points P1 - P4 in Figure 6. The reduced test case set now includes $TS = 11 \cdot 4 = 44$ combinations. Using (4), the number of fault sequences is calculated by $TD = 44 \cdot 43 = 1892$.

If the test shows that the PRO 'tripping characteristic' gives the correct output signal for all combinations of equivalence classes, it is possible to delete further test cases from the test case set. If it is shown once that the output variable has always the value TRUE for all three test points P2, P3 and P4 it suffices for the further test serie only to test one of the three test points, because the other ones don't give additional information. So the number of test points is reduced to a half. The values TS and TD result in $TS = 22$ and $TD = 22 \cdot 21 = 462$.

Looking at the remaining test case combinations, their number may be reduced once more, using informations about the phase-cycle symmetry. It can be shown that the number of fault combinations may be decreased to $TD = 54$. If this test case set is still too large, there is the possibility to filter some special fault types of the test case set like evolving faults or cross-country faults. As shown in this example the number of test cases TS was reduced from 256 to 22 and the number of test cases TD was reduced from 65280 to 54 using effective reduction techniques. This is equivalent to a reduction of the single test cases to 9 % and of the test case combinations to 1 % refered to the original test case set.

IX. SUMMARY

The probability of appearance of bugs in a software system increases with its size and complexity. Using a dynamical equivalence class test the demanded behaviour of the protection relay is verified. Because of the large number of test cases the usage of effective reduction techniques is required. The support of the expert-knowledge which is stored in the reduction algorithm includes a considerable potential to save test cases. Using a computer controlled protection test device where the test cases are simulated in a minimal test network, the automated run of testing a protection relay is possible.

X. REFERENCES

[1] AEG Guide Book: *Distance Protection Relay PD531*
[2] B. Beizer: *Software Testing Techniques*; Van Nostrand Reinhold, New York, 1990
[3] *ATP-Rule book*; Leuven/Belgium, Leuven EMTP Center, 1992
[4] B. Garrett, H. Dommel, K. Engelhardt: Digital Simulation of Protection Systems under Transient Conditions; *Proc. of the 9th Power Systems Computation Conference*; pp. 291-297, Cascais, Portugal, 1987,
[5] P. Liggesmeyer, H.M. Sneed: *Testen, Analysieren und Verifizieren von Software*; Springer Verlag, Berlin, Heidelberg, 1992
[6] M. Lipow: Number of faults per line of code; *IEEE Trans. Soft. Eng.*, 8/92, pp. 437-439
[7] M. Kezunovic, Y. Xia, C. Fromen, D. Sevcik: Distance Relay Application Testing using a Digital Simulator; *IEEE Trans. Pow. Del.*, vol.12, no.1,1997, pp. 72-82
[8] D. Nelles, H. Opperskalski: *Digitaler Distanzschutz*; Deutscher Universitäts-Verlag GmbH, Wiesbaden, 1991
[9]. D. Nelles, H. Opperskalski: Error Insensitive Distance Protection, *ETEP* 2 (1992), no. 4, pp. 245-251
[10] N. Parrington, M. Roper: *Softwaretest*; McGraw-Hill, Hamburg, New York, 1990
[11] R. Simon: *Netzbedingte Unschärfen des Distanzschutzes*; Dissertation D386, Universität Kaiserslautern, 1994