Evaluation of DSP based Numerical Relay for Overcurrent Protection

Yin Lee Goh, Agileswari K. Ramasamy, Farrukh Hafiz Nagi, Aidil Azwin Zainul Abidin
Centre for Communication Service Convergence Technologies (CCSCT)
Department of Electronics and Communication Engineering
Universiti Tenaga Nasional
Jalan IKRAM-UNITEN, 43000 Kajang, Selangor
MALAYSIA
yin_lee@hotmail.com, agileswari@uniten.edu.my, farrukh@uniten.edu.my, aidilazwin@uniten.edu.my

***Abstract*—**The reliability and security electrical supply is an important factor in modern society. However, the increasing complexity of power systems makes it difficult for protection operation to achieve these objectives. Nevertheless, numerical relays embedded with digital signal processor (DSP) are able to improve the protection operation significantly. The relays are capable of performing complex processing faster and with higher accuracy since the processing using DSP are optimized for real-time signal processing. In this paper, an overcurrent relay is built and investigated using DSP, TMS320F2812. The overcurrent protection is chosen since it is used as a major protection in the distribution systems. The overcurrent relay is modeled in MATLAB/Simulink before it is implemented on the DSP. Comparison results between simulation and hardware execution based on two implementation methods are presented. The performance evaluation of the relay in terms of operation time, memory capacity usage, execution time and transient analysis is investigated.

***Keywords*—**Power system protection, protective relaying, overcurrent protection, digital signal processors

I. Introduction

PROTECTIVE relays greatly impact power systems. Protective relays are used to detect any abnormalities in a power system and isolate the faulty part of the system in the shortest time. Protective relays are designed to maintain high degree of service continuity and limit equipment damage in the power systems. Severe disruption to the normal routine of modern society such as power outages is likely to increase the emphasis on reliability and security of supply electrical energy to consumers [1].

The expanding of power systems such as intense increase of transmission line capacity and increase of grids looping degree will increase the complexity of power system. This will cause the protection operation to become more difficult [2]. The protective relays encounter several problems as revealed in [2]. The present protection technique used is unable to ensure selectivity and speedily operation for the faults appearing on the entire protected line.

Numerical relays are able to improve the performance of the protection operation considerably. Numerical relay are embedded with specialized digital signal processor (DSP) as the computational hardware [1]. By using DSP as the relay's processor, the relay is capable of meeting the fundamental protective requirements such as reliability, sensitivity, selectivity and speed [3]–[4]. Therefore, the use of numerical relays will soon replace previous relays technology such as digital relays, static relays or even electromechanical relays.

The processing of numerical relays using DSP is also optimized for real-time signal processing applications [1]. As a result, the relay is capable of performing complex digital signal processing to detect fault faster and more accurately compared to previous relay technology. As a result, the reliability of the relay also increases. Furthermore, the usage of the numerical relays is not limited for protection only. This high performance relays is also enabled with advanced communication, self-supervision, ability to control, metering and also event recording [2], [5].

Numerical relays provide a wide range of protection functions such as overcurrent, directional overcurrent, undervoltage, overvoltage and also other types of protection [6]. Overcurrent protection is considered as the backbone of any protection strategy especially in distribution systems [7]–[8]. Distribution systems being the largest portion of the power system network, therefore the diagnosis of faults in this system is a challenging task. The faults that occur in distribution systems will affect the power system reliability, security and quality [9].

For the overcurrent protection, the relay operates with or without an intended time delay and trips the associated circuit breakers when the current flowing into the relay exceeds a setpoint value [10]. Overcurrent occurs due to presence of faults or overload conditions. For the faults caused by short circuit, the current present may be many times larger than its normal value. Meanwhile, overloads occur if the current exceeding the rated value. These phenomena can cause serious problems because the present of large amount of current may have severe damage to both the faulty part and healthy part of power system [11].

Even a short duration of transients in voltage and current may affect the operation of the protection relays. As a result, this will cause the relays to fail-to-trip or mal-trip event to occur [12]. Fail-to-trip occurs when the relay fails to trip in the presence of faults. Mal-trip occurs when the relay trips even though it is in healthy condition. Thus, in this paper a high speed improved protection for transient faults by using numerical relay is implemented and its performance is examined.

The overcurrent relay is built using numerical relay technology such that the processor used in the relay is a DSP. A DSP from Texas Instruments, TMS320F2812 is chosen to build the overcurrent relay according to inverse definite minimum time (IDMT) characteristics. TMS320F2812 is a high performance processor which is operating at frequency of 150MHz [13]. The improved Harvard architecture of TMS320F2812 enables processing of high amount of data. This processor provides high speed precision calculation for the mathematical algorithm involved in the processor [14]. These features are very suitable to implement the overcurrent relay for control applications.

In this paper, the performance of the implemented overcurrent relay on that DSP is evaluated. The relay's operation time is essential to be evaluated since it is used to determine the time for the relay to start to operate to trip the circuit breakers. The time delay can be critical for protection selectivity and system stability [15]. Thus, the process to obtain the relay's operation time should be performed as fast as possible. The time taken by the relay to obtain the output is also known as execution time. The execution time is examined along with the usage of memory capacity by the protection algorithm. Lastly, the relay is tested under transient circumstances, where a transient input is supplied into the relay. This is to ensure that the relay is able to detect transient faults.

In this paper, the implementation of overcurrent relay using TMS320F2812 is described. The overcurrent relay is modeled in MATLAB/Simulink before it is implemented on a DSP. Comparison results between simulation in MATLAB/Simulink and execution on TMS320F2812 based on two implementation methods are presented. For the first method, the simulation model from MATLAB/Simulink is directly downloaded into the DSP whereas for the second method, code in C programming is written on the DSP to represent the relay. The performance evaluation of the relay is in terms of operation time, memory capacity usage, execution time and transient analysis.

II. Digital Signal Processor (DSP)

DSP is a specific microprocessor that is designed for typical mathematical operations to manipulate measured digital data. DSP is capable of processing data speedily and generate output data in real-time. The additional hardware units embedded in the processor will speed up the computational of sophisticated mathematical operations [16]. Hence, this will reduce the memory capacity and number of execution cycles in the processor. As a result, the DSP operates faster.

Texas Instruments is a leading company producing DSPs. There are three types of DSPs families depending on its application. C2000 family is efficient function for real-time control applications and C5000 family is focused on mobile system with very efficient power consumption. Lastly, C6000 family is used for image processing, audio and communication applications [16]–[17].

Since, the C2000 family is competent for real-time control application, it is chosen to implement overcurrent relay as discussed in this paper. The processor selected is TMS320F2812. TMS320F2812 is a high speed fixed-point processor operates at 150MHz. This processor is also equipped large memory capacity of 18K words on-chip RAM and 64K words off-chip SRAM memory that sufficient to store large program [13].

Furthermore, the processor is also equipped with 12-bit analog-to-digital converter (ADC) core with built-in dual sample-and-hold (S/H). The ADC has fast conversion time which runs at 25MHz. With the autosequencing capability, the ADC provides up to 16 autoconversions in a single session [18]. All these features are very suitable for overcurrent relay implementation.

III. Characteristics Of Overcurrent Relay

The IDMT overcurrent relay is used as the primary scheme to protect power systems [19]. For IDMT characteristics, the relay starts to operate after an intended time delay. The time delay is also known as operation time. The advantage of the IDMT characteristics is that the greater the fault currents, the shorter are their operating time [20].

According to IEC 255-3 standard which is used by some of the protective relay manufacturer [21]–[25], the characteristics of IDMT relays are represented as in (1). Different types of inverse characteristics such as standard inverse, very inverse, extremely inverse and long inverse can be obtained by varying α and C values.

$$t = \frac{C}{\left(\frac{I}{I_S}\right)^{\alpha} - 1} \times TMS \tag{1}$$

where
- t - relay operation time
- C - constant for relay characteristics
- TMS - time multiplier setting
- I - current detected by relay, $I > I_S$
- I_S - current setpoint
- α - constant representing inverse time type, $\alpha > 0$

IV. Overcurrent Relay Simulation Model

The overcurrent relay is modeled in MATLAB/Simulink before it is implemented on a DSP as shown in Fig. 1. The IDMT relay is built based on IEC 255-3 standard as described in previous section. Different types of inverse characteristics which are included in the relay simulation model are standard inverse, very inverse and extremely inverse.

In the simulation model, the input currents are supplied with sine waveform of 50Hz. Different input values are supplied in

TABLE I.
PARAMETERS FOR DIFFERENT TYPES OF INVERSE CHARACTERISTICS

Relay Characteristics Type	α	C
Standard inverse	0.02	0.14
Very inverse	1	13.5
Extremely inverse	2	80
Long inverse	1	120

order to evaluate the performance of the relay under different circumstances. The inputs are quantized because the operation of a numerical relay is processed in digital form.

The input currents are sampled at sampling frequency of 1kHz [19]. The sampling frequency must be at least twice of the fundamental frequency. This is to ensure that the Nyquist criterion is fulfilled so that the aliasing of the input signals are avoided. With this sampling rate, 20 input samples per cycle are obtained.

These samples are used to compute root mean square (RMS) of the input current. The RMS computation is necessary in order to extract the fundamental component of the input current samples.

Next, these calculated RMS currents are supplied into the relay for evaluation and decision making inside relay [26]. These values are used by relay protection algorithm to determine the operation time.

The flow chart of the protection algorithm for overcurrent relay is shown in Fig. 2. The protection algorithm requires current setpoint value, TMS and type of inverse characteristics to determine the operation time. In this simulation model, very inverse characteristic of IDMT type with TMS of 0.1 is used to investigate performance of the relay.

The calculated RMS values are then compared with current setpoint value to obtain the current ratios. In this proposed model, the current setpoint is set as 150A. If the current ratio is more than 1, this indicates that the RMS input current exceeded the setpoint value. Hence, the protection algorithm embedded in the relay starts the process to determine the operation time.

The operation time is calculated accordingly based on inverse characteristic of the IEC standard, refer to (1). The operation time obtained is compared with a timer. The relay will trip if the operation time exceeds the timer and an output tripping signal is generated.

The simulation model of IDMT overcurrent relay is used to implement on the DSP, TMS320F2812. The performance of overcurrent protection on the DSP is examined. Comparison results between the MATLAB/Simulink simulation and hardware implementation are presented.

V. IMPLEMENTATION METHODS

There are two types of implementation methods used to implement IDMT overcurrent relay on the DSP, TMS320F2812. For the first method, the overcurrent relay simulation model from MATLAB/Simulink is directly downloaded into the DSP. Meanwhile, for the second method, code in C is written on the DSP to represent the relay.

A. Downloaded Simulation Model

In the first implementation method, the simulation model from Fig. 1 which was simulated in MATLAB/Simulink is directly downloaded into the DSP. The relay simulation model which uses Simulink blocks is implemented on the DSP. The Simulink model is converted into a compatible code so that it could be read by the processor and executed in real-time implementation. The integration between MATLAB and Texas Instruments will automate the process of generating the compatible code for the processor. Then, the generated code is complied, linked, downloaded and executed on the processor [27].

In order to view the results from the execution using this implementation, the simulation parameters are stored in the memory of the processor. Each parameter is stored in different memory location of the processor. The stored parameters are RMS value, current ratio and operation time. The stored parameter results are automatically updated while the processor is running. After the execution, the results can be viewed from the particular memory locations from an interface developed by Texas Instruments called as Code Composer Studio (CCS) Integrated Development Environment (IDE).

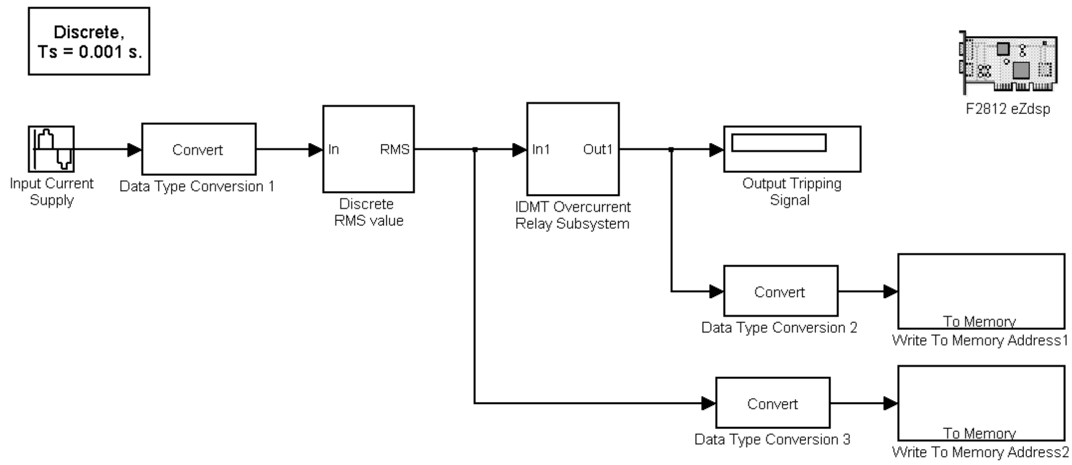

Fig. 1. The simulation model of overcurrent relay

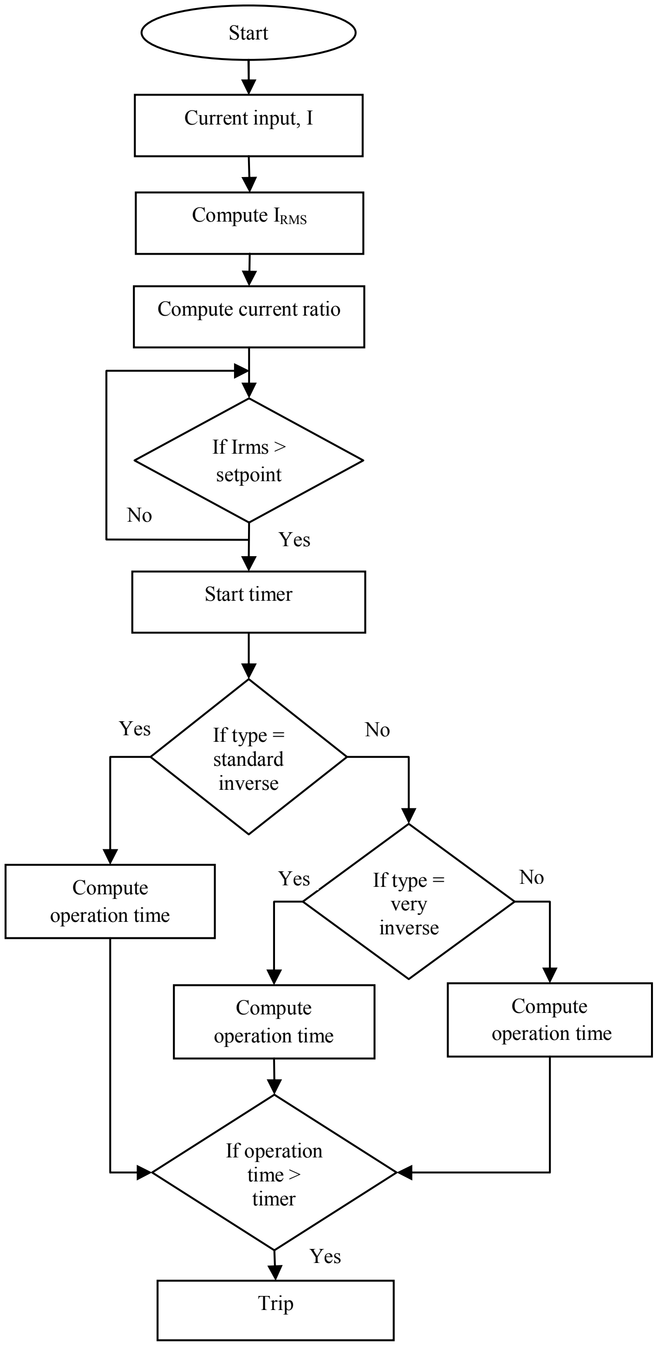

Fig. 2. Flow chart of the protection algorithm for overcurrent relay

B. Written C Code

For the second method, the overcurrent relay model is written in C code rather than downloading the simulation model from MATLAB/Simulink into the DSP. The code is written such that it has same functionality as in the simulation model.

In the written code, the sine waveform for input current of 50Hz is generated. The input currents are also sampled at sampling rate of 1kHz. Then, these current samples are buffered in the processor before proceed to compute RMS values. This is to ensure that the DSP has accurate number of input data per cycle to compute the RMS current. The RMS computation used in the written C code as in (2).

$$RMS = \sqrt{\frac{1}{n} \sum_{n=0}^{n} I^2} \qquad (2)$$

where n - number of samples
I - current value

The RMS values are used to calculate the current ratio before proceed to determine the operation time. The same protection algorithm from simulation model is used. Once the relay setting is selected, the relay will trips if the RMS input current exceeds the current setpoint value and after the operation time is satisfied.

In order to ensure that the relay is operating correctly, the output tripping signal is buffered in the processor. The results of the RMS value, current ratio, operation time and output tripping signal are also viewed from CCS.

Comparison results for the relay operation time between MATLAB/Simulink simulation and execution on the DSP for two different methods are performed. Others performance analysis such as memory capacity usage and execution time for both implementation methods are presented. Lastly, the transient performance testing is performed for the written C code only.

VI. HARDWARE IMPLEMENTATION

The implementation for the overcurrent relay consists of DSP, TMS320F2812 and a computer as the DSP host. The block diagram of the hardware implementation is shown in Fig. 3. The DSP of TMS320F2812 is built on a board called as

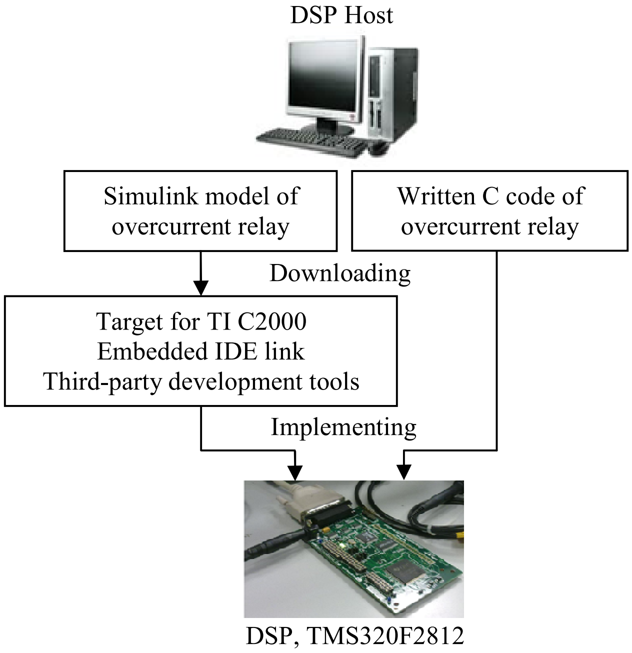

Fig. 3. Hardware implementation of overcurrent relay on TMS320F2812

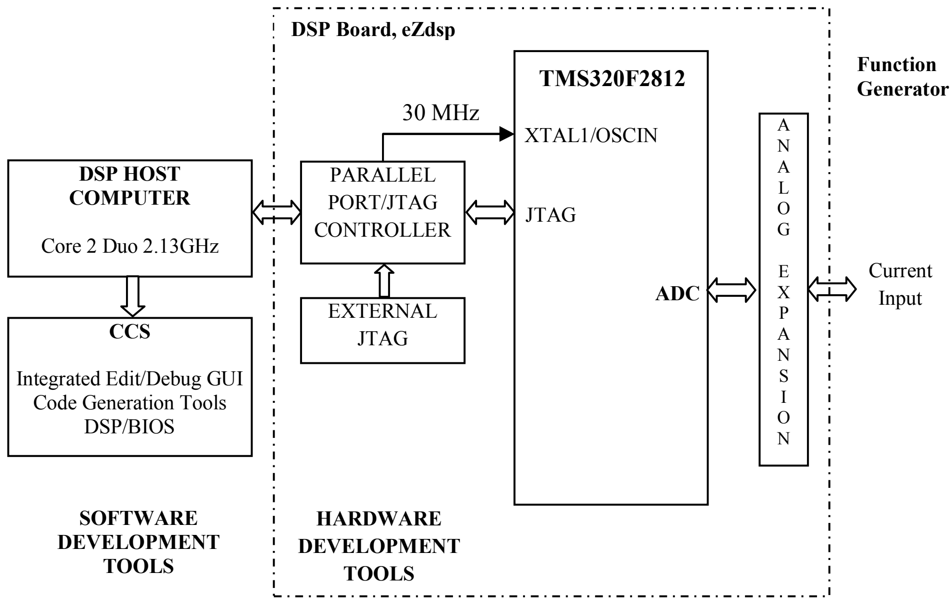

Fig. 4. The block diagram of the basic configuration for eZdsp TMS320F2812

eZdsp TMS320F2812. The eZdsp TMS320F2812 is a stand-alone board that allows full speed verification of TMS320F2812 code [28]. The block diagram of the basic configuration for eZdsp TMS320F2812 is shown on Fig. 4 [28].

There are two types of methods used to implement the overcurrent relay on the DSP. In the first method, the simulation model is directly downloaded into the DSP. The simulation model is required to be converted into a compatible code that could read by the processor. This is achieved with incorporation of Texas Instruments and MATLAB to produces specialize MATLAB's toolboxes for the processor. The toolboxes are Target for TI C2000 and Link for Code Composer Studio [27]. The combination of Target for TI C2000 and Real-Time Workshop enables CCS to act as an integrated environment which does not require coding.

The executing code is automatically generated from Real-Time Workshop which used by Target for TI C2000. Then, the Texas Instruments development tools generate a C language real-time implementation of the Simulink model. The Real-Time Workshop also builds a CCS project from the C code. The generated source code must be compiled and linked by using CCS so that it can be loaded and executed on the DSP. The Target for TI C2000 uses Link for Code Composer Studio to start the code building process within CCS [27].

Meanwhile for second method, code in C is written on the DSP to represent the relay. The written code in the second method is required to be written in the CCS. Then, the written code is also complied, linked, downloaded and executed on the processor using CCS. After downloaded the executable code on the processor, the code runs wholly on the DSP.

Either the code is generated automatically by the simulation model or by written, this code is downloaded into the processor through the DSP host using parallel port connector. The eZdsp TMS320F2812 is implanted with a parallel port interface to enable process for loading the code from CCS into the processor. The processor is hosted on a Core 2 Duo 2.13GHz computer. All the results from the DSP execution for both implementations are viewed from CCS either in the form of graph or value.

VII. Results

The results obtained from MATLAB/Simulink simulation and execution on the DSP for the overcurrent relay is described in this section. The performance parameters used to investigate the implemented relay are presented. The performance parameters are operation time, memory usage, execution time and transient analysis.

A. Operation Time

The operation time of the overcurrent relay is the time delay that the relay should fulfill before it trips the circuit breakers in the distribution system. The relay is required to trip at right timing to avoid any fail-to-trip or mal-trip event. In order to investigate the accuracy of operation time obtained from simulation and also different implementation methods on the DSP, different input current values are supplied into the overcurrent relay. The results from MATLAB/Simulink simulation and execution on DSP for two different methods are shown in Table II.

From Table II, the operation time is obtained for very inverse type characteristics of the IDMT overcurrent relay with TMS of 0.1. The current setpoint of the relay is set as 150A. The relay will start to compute the operation time if the RMS input current exceeds the setpoint value. Thus, the processor must be able to operate efficiently in the real-time to detect the input current value. Failure to detect the fault current is likely to cause damage to the equipments and place the entire plant in danger.

TABLE II.
OPERATION TIME OF THE OVERCURRENT RELAY FROM MATLAB/SIMULINK SIMULATION AND EXECUTION ON DSP

Amplitude input current (A)	Operation time (s)			
	According to IEC standard	Simulation from MATLAB/Simulink	Execution on DSP	
			Downloaded simulation model	Written C code
170	-	-	-	-
210	-	-	-	-
213	329.9419	329.9	324.6737	329.9426
227	19.2614	19.26	19.34454	19.26133
500	0.9948	0.9948	0.994825	0.9948252

The operation time obtained from the simulation and hardware implementations have small variation as shown in Table II. The written C code which is embedded in the processor produces closely same results as in the simulation and according to IEC standard. The operation time obtained from written C code has higher accuracy compared to downloaded simulation model.

Since the overcurrent relay will only compute the operation time when the RMS input exceeds the setpoint value thus there are no results obtained for the RMS input current less than the setpoint value. These are shown for the amplitude input current of 170 A and 210A.

For the amplitude input current of 213A is the minimum value that could cause the relay to trip since the ratio of this input value is 1.004. With this minimum input current value, the theoretical value obtained for the operation time is 329.9419s. Compared to MATLAB/Simulink simulation is 329.9s and execution on the DSP using downloaded simulation model and written C code is 324.6737s and 329.9426s respectively.

B. Memory Capacity

The memory capacity used by the implemented overcurrent relay on the DSP for both types of implementation methods is investigated. The memory usage is evaluated in order to examine the effect of memory capacity usage compared to execution time used by the protection algorithm of overcurrent relay.

The memory capacity of the 32-bit processor used in the hardware implementation shows a large difference between downloaded simulation model and written C code. The downloaded simulation model into DSP requires large amount of memory usage to store the automatically generated code by MATLAB. The memory capacity used by the downloaded model is 3639 addresses and written C code is 2681 addresses. By writing C code for the relay, small memory allocation is used to store the program in the processor.

C. Execution Time

The execution time is the time taken by the implemented overcurrent relay on the DSP to execute the protection algorithm in real-time to obtain the operation time. The execution time taken by both implementation methods is obtained.

The execution time is obtained by using General Purpose Input Output (GPIO) port of the processor. A GPIO pin is used to toggle at the beginning and at the end of the protection algorithm. Then, the GPIO is used to observe the time period of positive pulse width using an oscilloscope. The time period of positive pulse is the execution time used by protection algorithm.

As shown in Table III is the execution time obtained for both implementation methods. By using written C code for the algorithm, the execution time obtained is 740µs compared to downloaded simulation model is 50µs. The execution time obtained is shorter by downloading the simulation model. This is due to the automatically generated code is optimized for real-time implementation, thus it has faster operation time compared to the written C code.

The written C code has longer execution time because the RMS computation took 720µs to complete computation for an input cycle. The execution time of the RMS computation for written C code is shown in Fig. 5. Almost 97% of the execution time is contributed in the RMS computation. This is

TABLE III.
EXECUTION TIME FOR THE IMPLEMENTED OVERCURRENT RELAY ON DSP

Implementation methods	Execution time (s)
Downloaded simulation model	50µ
Using C-Language written C code	740µ

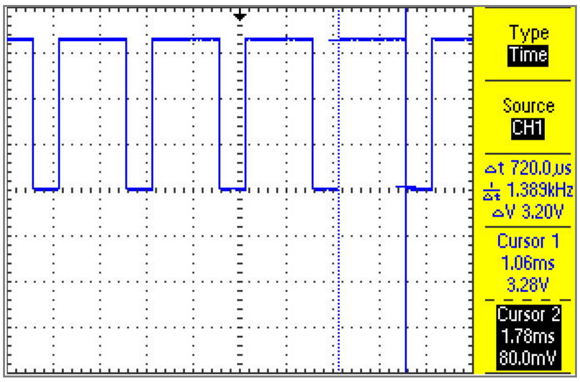

Fig. 5. Execution time of the RMS computation for written C code

caused by the input current which is generated in the program itself. Since, the input is required to be buffered in the processor to obtain 20 samples per cycle before perform the RMS computation. Thus, there is a looping process for 20 times and this cause more time for the written C code method.

Overall, the downloaded simulation model method uses larger memory capacity but execution time is shorter compared to written C code which uses less memory capacity but execution time is longer. However, the operation time obtained from written C code has higher accuracy compared to the downloaded simulation model. A slightly change of the input current value will affect the operation time drastically.

D. Transient Testing

Transient currents that are generated due to non-fault events should not lead to trips the relay. In order to avoid this, the overcurrent relay must be able to detect and process the information of the input efficiently. The transient testing is performed on the implemented relay to test the capability of the DSP to detect sudden change of input current so that fail-to-trip or mal-trip event is avoided.

In this testing, the transient waveform is supplied into the overcurrent relay rather than fixed programmed amplitude waveform. In the testing, a transient input current is injected into the relay for the written C code implementation. This is because by using C code program, a real-time testing is able to be carried out by supplying transient input from function generator into the DSP.

The DSP is embedded with an ADC. Hence, this will ease the process of converting the inputs into digital before it is processed in the processor. By using this internal ADC, it will eliminate the time delay caused by using external ADC during transferring data from the ADC to DSP. Since, the embedded ADC on the processor is utilized, slight modification on C code program for the input part is required. The ADC is initialized in order to employ the ADC peripheral on the processor.

The input current is scaled down before supplied into the processor. This is due to the analog input supplied into the ADC must be in the range between 0-3V. If the analog input is exceeded, the ADC on the processor will spoil. Thus, the input is scaled down such that the maximum input current is 1273A (RMS*√2) corresponding to 2.6V peak-to-peak value is assumed as it is the acceptable value for the protection algorithm of an overcurrent relay.

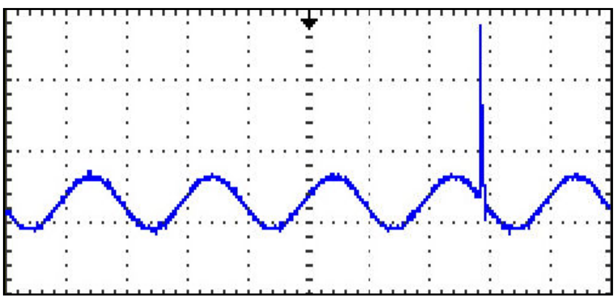

Fig. 6. Transient waveform created by external circuit

The transient waveform is produced from an external circuit that triggers a transient to the input supply from the function generator when activated. The external circuit consists of capacitor which will discharge charges when activated and transient is generated.

In the testing, input current is initially supplied with amplitude of 188A and current setpoint for the relay as 161A. Since the transient is randomly generated, thus different transient values are obtained each time. Fig. 6 shows the transient waveform obtained with maximum value of 395A.

The result of transient testing is shown in Fig. 7 and Table IV. Table IV shows the values for current ratio and trip value before and after transient. From the results obtained, the implemented relay proves its capability to detect the transient waveform. The transient occurs during the input current cycle of 2 and 106 and as a result, the RMS input current exceeds the setpoint value, hence the trip value becomes 0. In the next current cycle, when the input current changes back to steady conditions, the trip value becomes 1 again as shown in Fig. 7.

The DSP is able to detect the transient waveform each time it is triggered. Since, the transient generated is not a fault event it will not cause the relay to trip as observed from the experimental testing. This transient testing shows that the implemented overcurrent relay on the DSP is proficient to detect transient waveform and able to process the input current waveform correctly and efficiently to avoid fail-to-trip or mal-trip to occur.

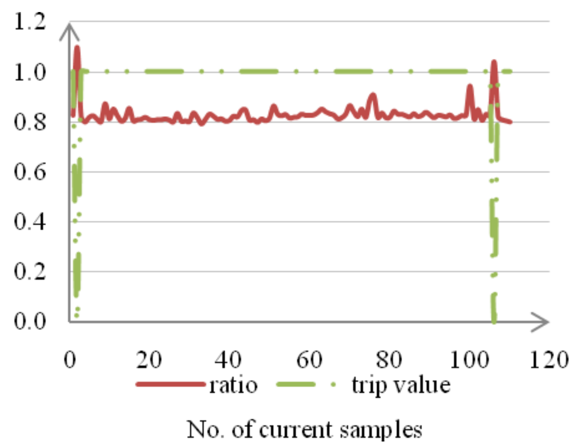

Fig. 7. Current ratio and trip value and for transient testing

TABLE IV.
CURRENT RATIO AND TRIP VALUE FOR PARTIPULAR INPUT CYCLES

No. of input current cycles	Current ratio	Trip value
1	0.8277733	1
2	1.1015990	0
3	0.8214784	1
104	0.8324944	1
105	0.8277733	1
106	1.0433720	0
107	0.8230520	1
108	0.8104624	1

VIII. CONCLUSIONS

This paper describes the performance evaluation of protective relay using DSP, TMS320F2812 for overcurrent protection. Results clearly indicate that the operation time obtained for both implementations methods are similar to IEC 255-3 standard. However, the downloaded simulation model method uses larger memory but its execution time is shorter compared to the written C code method which uses less memory but its execution time is longer. For transient analysis, the relay is proficient in detecting transient input. The DSP based overcurrent relay using both implementation methods are proven to provide adequate reliability and security with improved performance. In future, the method selected to implement the overcurrent relay will depend on the preference of memory usage or execution time of the protection algorithm. Future works include implementing the relay on the DSP using artificial intelligence.

REFERENCES

[1] *Network Protection & Automation Guide*, Areva T&D, 1995.
[2] A. Iagar, G. N. Popa, C. M. Dinis, and G. Moraru, “Study about numerical relay SEL-387 for overcurrent and differential protections of 110/20 kV transformers,” in *Proc. of the 13th WSEAS Int. Conf. on Systems*, Greece, 2009, pp. 265–270.
[3] E. Price, “The next step in the evolution of protection and control implementation,” in *Annual Conference for Protective Relay Engineers*, 2010, pp. 1–16.
[4] M. Khederzadeh, “Back-up protection of distance relay second zone by directional overcurrent relays with combined curves,” in *IEEE Power Engineering Society General Meeting*, 2006.
[5] H. A. Darwish, and M. Fikri, “Practical considerations for recursive DFT implementation in numerical relays,” *IEEE Trans. on Power Delivery*, vol. 22, pp. 42–49, January 2007.
[6] W. A. Elmore, *Protective Relaying Theory and Applications*, ABB Power T&D Company Inc, 1994.
[7] P. P. Bedekar, S. R. Bhide, and V. S. Kale, “Optimum time coordination of overcurrent relays in distribution system using Big-M (Penalty) method,” *WSEAS Trans. on Power Systems,* 4(11), pp. 341–350, November 2009.
[8] D. N. Vishwakarma, and Z. Moravej, “ANN based directional overcurrent relay,” *IEEE/PES Transmission and Distribution Conf. and Exposition*,” vol. 1, 28 October–2 November 2001, pp. 29–64.
[9] W. Al-Hasawi, and M. Gilany, “Proposed techniques for identifying open and short circuit sections in distribution networks,” *WSEAS Trans. on Power Systems*, 4(12), pp. 372- 381, December 2009.
[10] J. C. Tan, P. G. McLaren, R. P. Jayasinghe, and P. L. Wilson, “Software model for inverse time overcurrent relays incorporating IEC and IEEE standard curves,” in *Canadian Conf. on Electrical and Computer Engineering*, vol. 1, 12–15 May 2002, pp. 37–41.
[11] R. Lin, “Development of protective relaying equipment in substations,” *Proc. of the 2006 IASME/WSEAS Int. Conf. on Energy & Environmental Systems*, Chalkida, Greece, May 8-10, 2006, pp. 373–378.
[12] F. Wang, and M. H. J. Bollen, Classification of component switching transients in the viewpoint of protective relays,” *Electric Power Systems Research*, 64(3), pp. 197–207, March 2003.
[13] *TMS320F2810, TMS320F2811, TMS320F2812, TMS320C2810, TMS320C2811, TMS320C2812 Digital Signal Processors Data Manual*, Texas Instruments, 2010.
[14] K. Shuang, C. H. Chen, N. Xiao and D. Z. Yu, “A new digital electronic governor based on 32 bits DSP for a gas engine”, in *9th Int. Conf. on Control, Automation, Robotics and Vision, 5–8 December 2006*, pp. 1–6.
[15] F. Muzi, “A filtering procedure based on least squares and Kalman algorithm for parameter estimate in distance protection.” *International Journal of Circuits, Systems and Signal Processing*, 1(1), pp. 16–21, 2007.
[16] *C2000 MCU Teaching ROM, TMS320F2812 Digital Signal Processor Implementation Tutorial*, Texas Instruments.
[17] N. Dahnoun, *C6000 DSP Teaching ROM*, Texas Instruments, 2004.
[18] *TMS320x281x DSP Analog-to-Digital Converter (ADC) Reference Guide*, Texas Instruments, 2005.
[19] D. Campos, E. Moreno, and D. Torres, “Test and evaluation time-inverse over-current protection algorithm using SIMULINK,” *Proc. of the 7th WSEAS Int. Conf. on Signal Processing,* Istanbul, Turkey, 2008, pp. 69–74.
[20] Computer representation of overcurrent relay characteristics, *IEEE Power Engineering Review*, vol. 9, pp. 50–51, July 1989.
[21] Yao-Hung Chan, Chi-Jui Wu, Wei-Neng Chang, and Ying-Pin Chang, “Voltage sag planning of industry power system using hybrid differential evolution considering CBEMA curve,” *International Journal of Circuits, Systems and Signal Processing*, 5(2), pp. 141–150, 2010.
[22] N. X. Tung, G. Fujita, M. A. S. Masoum, and S. M. Islam, “Impact of harmonics on tripping time and coordination of overcurrent relay,” in *7th WSEAS Int. Conf. on Electric Power Systems*, *High Voltages, Electric Machines*, Venice, Italy, November 2007, pp. 46–52.
[23] *Single Input Energizing Quality Measuring Relays with Dependent or Independent*, IEC Publication 255-3 (1989–05), 1989.
[24] Jarm-Long Chung, Ying Lu, Wen-Shiow Kao and Chih-Ju Chou, “Study of solving the coordination curve intersection of inverse-time overcurrent relays in subtransmission systems,” *IEEE Trans. on Power Delivery*, vol. 23, no. 4, pp. 1780–1788, 2008.
[25] A. A. Zainul Abidin, A. Ramasamy, I. Z. Abidin, and F. H. Nagi, “Overcurrent time delay determination using gain scheduled PID controllers,” *3rd Int. Conf. on Energy and Environment*, 2009, pp. 89–93.
[26] N. Mikulandra, and M. Stojkov, “New challenges for protection system,” *Proceedings of the 4th IASME/WSEAS Int. Conf. on Energy & Environment*, February 2009, pp. 269–272.
[27] *Target for TI C2000 User's Guide*, The MathWorks, Inc., 2007.
[28] *eZdspTM F2812 Technical Reference*, Spectrum Digital, Inc., 2003.