A Numerical Relay Implementation for Overcurrent Protection Based on ARM Cortex – M4 Microprocessor

IOP Conf. Series: Materials Science and Engineering **752** (2020) 012005 doi:10.1088/1757-899X/752/1/012005

A Numerical Relay Implementation for Overcurrent Protection Based on ARM Cortex – M4 Microprocessor

Anh V Nguyen[1,a], Lien B Nguyen[1,b] and Anh Hoang[1,c]

[1]School of Electrical Engineering, Hanoi University of Science and Technology, Hanoi, Vietnam

[a]anh.nguyenvan1@hust.edu.vn; [b]Lien.nguyenbich@hust.edu.vn; [c]Anh.hoang@hust.edu.vn

Abstract. Numerical overcurrent protection relays are the most extensively used devices to safeguard the power system from detrimental faults which might lead to a catastrophe. This paper presents a practical implementation of a numerical overcurrent protection relay based on ARM Cortex – M4 microcontroller embedded with a floating-point unit which has ability to perform complex calculation operations with a great accuracy for real-time protection algorithms. In order to implement, the system is designed to capture analog values of input currents by using Hall current sensors, analog conditioning circuits and 12 – bit A/D converters. Based on the measured input current values, a real-time signal processing algorithm for overcurrent protection is applied to produce an inverse - time characteristic according to the IEEE Standard C37.112-1996. The experimental results demonstrate that our numerical relay produces great performances in the real-time and has ability to protect electrical components from exceeded current faults satisfying the IEEE standard for overcurrent relays.

1. Introduction

The most crucial purpose of a power system is to generate and distribute electric energy to consumers without interruptions. However, the power system is always vulnerable to faults due to either natural disasters or by misoperation of the system [1]. This can result in permanent damage to costly power system elements and disruptions of power supply to customers. In order to assist the power system to sustain faults as well as minimize the damage in critical components, overcurrent protection relays have been considerably utilized to protect the system from sudden rise in current.

An overcurrent relay uses inverse-time characteristics to determine tripping time to protect electrical elements from the power abnormal operating conditions. With this operating characteristic, the relay gives a shorter tripping time for high current magnitudes than that of low current magnitudes. There are two different types of relays: electromechanical relays and microprocessor-based or numerical relays. Compared to electromechanical relays, numerical protection ones are advantageous on the account that they provide better protection, reliability, troubleshooting, and fault information. This can explain a reason why electromechanical relays are gradually replaced by numerical ones in the power system [2, 3].

There are a number of reseachers who are interested in investing advanced inverse-time characteristics protection algorithms for numerical relays. Basically, there are two main methods: Direct interpolation from stored data [4, 5] and use of data-fitting formulas [6]. Direct interpolation

methods rely on storing tripping time data in the memory, and then selecting an appropriate tripping data point based on the ratio between an input current and a pickup current. This method is not accurate because most of the time, the actual tripping time is not match with the one stored in the memory, then an interpolation is neccessary to determine the approximated time. On the other hand, data fitting fomulas directly use the input current to determine a tripping time based on the inverse – time fomulus, thus, this method provides more accurate tripping time results for the protection relays.

This paper reports an implementation of a numerical relay for overcurrent protection based on ARM cortex – M4 microcontroller based on the data fitting formulas method. Embedded many advanced signal processing features, this 32-bit microcontroller is able to perform a complex protection algorithm efficiently as well as communicate with other devices easily. Besides, this paper also focuses on an overcurrent protection algorithm in order to produce an operating characteristic required by the IEEE Standard [7].The rest of the paper will be organized as follows. The section 2 presents a theory of overcurrent protection based on the IEEE Standard. Next, the section III provides an overview of the system including both hardware design and software design. Based on the section 2 and section 3, the section 4 details an overcurrent protection algorithm in order to achieve a desired inverse – time characteristic. Finally, experimental results and conclusions will be made in the section 5 and section 6.

2. Overcurrent Protection

Basically, an inverse – time current characteristic is defined by two regions: an operation (trip) region and a rest region. The operation region is applied for the condition where the input current (I) is greater than the pickup value (I_p). A relationship between tripping time (t (I)) and input current is determined by

$$\begin{cases} t(I) = \dfrac{A}{M^{p} - 1} + B \\ M = \dfrac{I}{I_p} > 1 \end{cases} \tag{1}$$

Where A, B and p are constants and selected for a specific inverse – time characteristic. The IEEE standard [7] specifies three types of characteristics: moderately inverse, very inverse, or extremely inverse corresponding to their constants shown in the table 1. The rest region, on the other hand, is defined for the condition where the input current is smaller than the pickup current. The tripping time in this region is calculated by

Table 1. Standard constants regarding to the types of the inverse time-current characteristic.

Characteristic	A	B	p	T_r
Moderately inverse	0.052	0.114	0.02	4.85
Very inverse	19.61	0.491	2	21.6
Extremely inverse	28.2	0.122	2	29.1

$$\begin{cases} t(I) = \dfrac{T_r}{M^{2} - 1} \\ M = \dfrac{I}{I_p}\ ;\ 0 < M < 1 \end{cases} \tag{2}$$

Where T_r is a constant depending on the type of the characteristic as shown in the table 1.

From the equation (1) and (2), it can be realize that the tripping time has positive values in the operating region while having negative values in the rest region. To combine both regions, the following dynamic equation must be satisfied

$$\int_0^{T_0} \frac{1}{t(I)} dt = 1 \tag{3}$$

Where T_0 is total tripping time when the current changes during the fault. The equation (3) implies that the tripping time for a time-varying input is equal to the harmonic mean of the delays relative to the current segments considered as constant over a very short time interval. In a discrete form, the above integral equation can take a form of a sum with integral step Δt as

$$\begin{cases} \sum_{i=0}^{N} \frac{\Delta t}{t_i(I)} = 1 \\ T_0 = N.\Delta t \end{cases} \tag{4}$$

3. System Overview

A completed hardware description of our developed numerical overcurrent relay is presented in the figure 1.

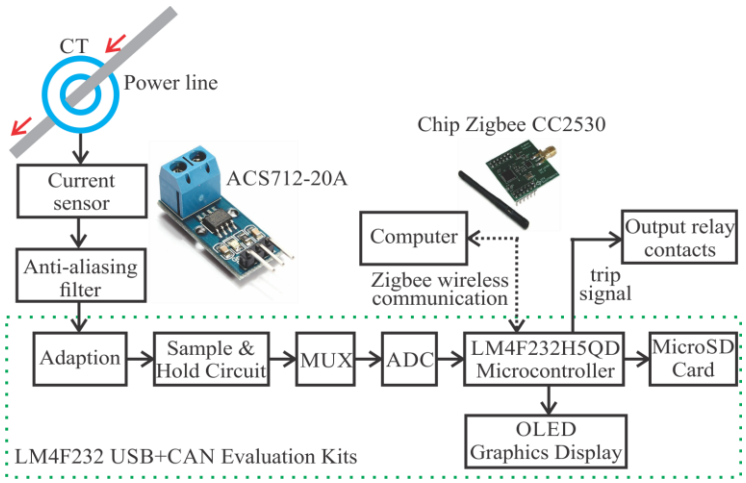

Figure 1. The numerical overcurrent relay hardware overview

The Allegro ACS 712 – 20A, based on the Hall Effect to sense either AC or DC current, are used to measure the input current. This sensor allows a linear conversion of the current margin [-20A, 20A] to the voltages [0.5V, 4.5V]. The voltage signal then is passed through an anti – aliasing second order Butterworth filter with a cutoff frequency of 500 Hz in order to attenuate electrical noises which have frequencies with 500 Hz or higher. It is necessary to avoid aliasing while sampling the input signals.

The center of this numerical relay is based on Stellaris LM4F232 evaluation board from Texas Instrument [8]. The figure 1 shows some parts which are already built in the evaluation board and utilized for this project. A purpose of the adaptation is to scale down the filtered voltage range into the 0 – 3V range of the 12 – bit ADC of the microcontroller. After receiving and processing signals from the ADCs, the microcontroller displays the results onto the 96 x 64 color OLED screen using SSP2 peripheral. In addition, it is also communicated with a host computer via either USB cable or Zigbee wireless communication using UART protocol. The relay obtains setting parameters from the computer for its overcurrent protected algorithm and send processed data to the computer for monitoring and recording. In a case of tripping, an output pin of the LM4F232H5QD chip emits a control signal to an electromechanical relay with optocoupler isolation in order to change the status of contacts which is used for a circuit breaker to disrupt the power supply circuit.

4. Algorithm

As discussed in the section 2, the inverse – time current characteristic of a relay is determined by the tripping time using the equation (1) and (2). However, before applying these equations, a RMS value of the

input current must be found first. Measuring a RMS value of a current can be done by either time domain algorithms or frequency domain algorithms. In this project, the method of average values [4] is applied in order to estimate the RMS value of the input current because this method consume less RAM memory and its convergence is much faster than others. When applied this method, the basic assumption is that the signals are sinusoidal. An average value of the current is calculated by

$$\begin{aligned} I_{avg} &= \frac{1}{T}\int_{t}^{t+T} \left|i(\mathrm{t})\right| dt = \frac{1}{T}\int_{t}^{t+T} \left|I_m \sin(\omega \mathrm{t}+\varphi)\right| dt \\ &= \frac{2}{T}\int_{t}^{t+T/2} \left|I_m \sin(\omega \mathrm{t}+\varphi)\right| dt = \frac{2I_m}{\pi} \end{aligned} \tag{5}$$

Where Im and T is a peak value and a period of the current.

Based on the equation (5) and the definition of the RMS value of the current, the relationship between the average value and RMS value of the current can be found as following

$$\begin{aligned} I_{RMS} &= \sqrt{\frac{1}{T}\int_{t}^{t+T} i^2(\mathrm{t})dt} = \sqrt{\frac{1}{T}\int_{t}^{t+T} I_m^2 \sin^2(\omega \mathrm{t}+\varphi)t} \\ &= \frac{I_m}{\sqrt{2}} = \frac{\pi I_{avg}}{2\sqrt{2}} \end{aligned} \tag{6}$$

From the equation (6), RMS value of the current can be easily determined by estimating its average value. Let's assume that there are m samples of the current i(k) that are captured by the A/D converter within a period T. Then, the equation (6) is represented in a discrete form as below

$$\begin{aligned} I_{RMS} &= \frac{\pi I_{avg}}{2\sqrt{2}} = \frac{\pi}{2\sqrt{2}}\frac{1}{m}\sum_{k=1}^{m}\left|i(k)\right| \\ &= \frac{\pi}{m\sqrt{2}}\sum_{k=1}^{m/2}\left|i(k)\right| \end{aligned} \tag{7}$$

The equation (7) implies that the average method allows us to find out the RMS value of the current during a time of half period with simple operations including only addition and multiplication.

After IRMS found, the tripping time can be easily obtained. However, to deal with the dynamics of the current change, the tripping condition described in the equation (4) must be met. In digital implementation, this can be done by using an accumulation function C(I)

$$C_k(I) = \frac{T_c}{t_k(I)} + C_{k-1}(\mathrm{I}) \tag{8}$$

Where subscript k indicates a loop kth of the program, and Tc is a time constant representing for a time interval between two consecutive loops of the program. A value of the constant Tc is selected by considering the sampling rate of A/D converter as well as computing speed of operators.

Based on the value of tripping condition Ck(I), the microcontroller determines whether tripping signals are sent to change the contact status or not. The value of $C_k(I)$ tends to increase when overcurrent is present, and decrease as the current is below the pickup value I_p.

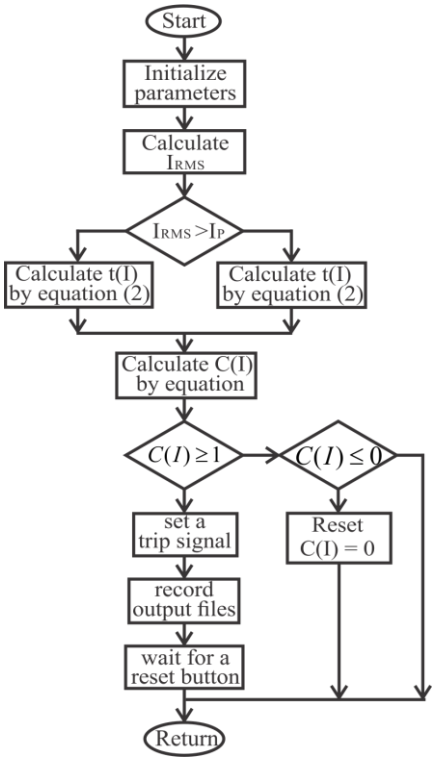

Figure 2. An algorithm for overcurrent protection

To summarize all above points, the figure 2 shows an overcurrent protection algorithm for our developed relay.

5. Experimental results

5.1 Setup

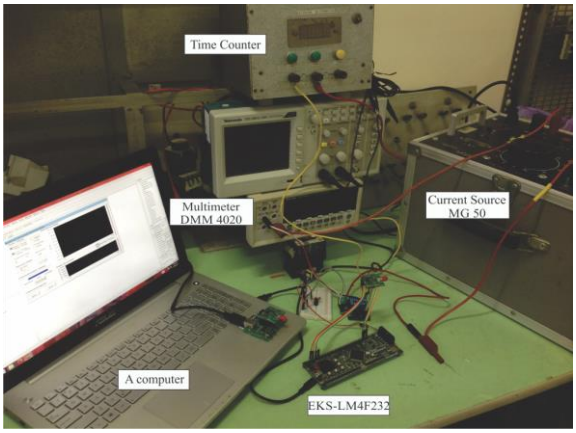

Figure 3. Experimental setup with unpacked parts of our numerical overcurrent relay

In order to investigate the performance of the developed system, several tests were conducted with our numerical overcurrent relay. The figure 3 presents an experimental setup for the evaluation tests. To generate test currents, we use a voltage – current set for protecting relay testing Cotel M50 which acts as a current source ranging up to 50A. A digital multimeter DMM 4020 of Tektronix is also recruited to measure the test current values during experiments. A computer is connected to a RF Zigbee DRF1605H module to wirelessly communicate with the relay. The user interface in the computer

enable a user to:

1) Control data acquisition speed via the timer setup.

2) View the measured current and tripping condition sent from the relay and record them into files as desired.

3)Change defined protection modes (moderate, very and extreme as described by the IEEE standard) or manually change parameters of the inverse – time characteristics.

We also tested the communication between the relay and the host computer via Zigbee wireless. We found that they were still able to communicate with each other very well in few hundred meters away. Finally, a time counter connected with output contacts of the relay is used to measure the tripping time.

5.2. Tripping time

Table 2. An accuracy of moderately inverse – time characteristic obtained by our developed relay

Test current (A)	Calculated trip time (s)	Relay average trip time (s)	Error (%)
0.85	-	-	-
1.05	52.87	52.59	0.53
1.5	6.44	6.34	1.55
2	3.80	3.77	0.79
2.5	2.90	2.85	1.68
3	2.43	2.40	1.32
5	1.69	1.71	1.28
7	1.41	1.42	0.59
9	1.26	1.28	1.56
10	1.21	1.19	1.39

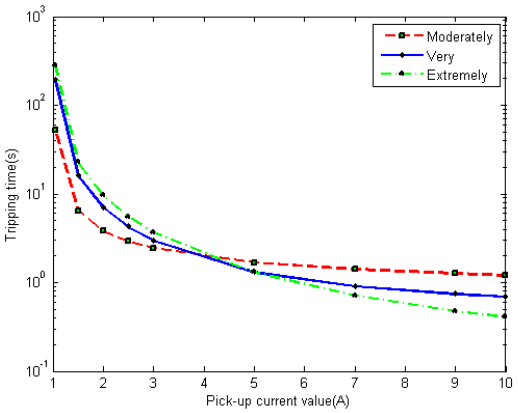

Figure 4. Inverse – time characteristics of our developed numerical overcurrent relay

We would like to measure the accuracy of the inverse – time characteristic and the difference between the protective characteristic types produced by our proposed relay. There are two experiments conducting for this purpose. The first experiment was carried out to measure an accuracy of moderately inverse - time characteristic of the developed system. In this experiment, the input current was varied to some preset values, and the tripping time of our numerical relay corresponding to a particular current value was recorded by the time counter. For the consistency, for each test current value, there are three repeated trials and the tripping time is averaged. Tripping time calculated by the

formulas of the IEEE Standard C37.112-1996 was also provided for a comparison purpose as tabulated in the table 2. From the table 2, it can be easily seen that the moderately inverse – time characteristic obtained from our system was highly accurate. When comparing with values determined from the IEEE standard, a percentage error is less than 2% which is highly acceptable.

The second experiment was to measure three types (moderately, very and extremely) of the inverse –time characteristic produced by our numerical overcurrent relay. The figure 4 shows these characteristics on semi-logarithm graph. From this plot, we can see that there were obvious differences between protection modes of our relay and all these characteristics were highly consistent to those of the IEEE protection standard.

6. Conclusion

This paper presents an implementation of a numerical overcurrent relay based on ARM Cortex M4 microcontroller which has advanced features such as Super Harvard Architecture and Floating Point Processing Unit. The system uses a Hall current sensor ACS 712 – 20A to convert the input current into a voltage signal which is filtered, conditioned, and then sampled by 12-bit high speed A/D converter. ARM Cortex M4 combined with powerful signal processing functionality effectively performs the average method to estimate RMS value of current and the overcurrent protection algorithm. The experiment results prove that the protection characteristics produced by our proposed system are highly accurate and very consistent with those of the IEEE protection standard. In the future, we would like to apply some advanced protection techniques such as artificial neural network in order to improve the performance of this system.

Acknowledgements

This research is funded by Hanoi University of Science and Technology (HUST) under grand number is T2018 – PC - 051

References

[1] L. Wenchen, S. Lu, Z. Weimei, and L. Chunqing, "Power system reliability analysis system based on PSASP and fault enumeration method and applications," in Electricity Distribution (CICED), 2014 China International Conference on, 2014, pp. 1323-1326.

[2] W. Li, L. Sun, W. Zou, and C. Luo, "Power system reliability analysis system based on PSASP and fault enumeration method and applications," in 2014 China International Conference on Electricity Distribution (CICED), 2014, pp. 1323-1326.

[3] S. Richards, D. Chatrefou, and A. Procopiou, "Movement to the full digital substation for primary distribution," in 22nd International Conference and Exhibition on Electricity Distribution (CIRED 2013), 2013, pp. 1-4.

[4] Z. N. Stojanovic and M. B. Djuric, "Table Based Algorithm for Inverse-Time Overcurrent Relay," Journal of Electrical Engineering, vol. 65, 01/21 2014.

[5] G. V. Kshirsagar, G. Mulay, and S. Yeolekar, TMS320F28335 based single phase overcurrent protection Implementation using Numerical Relay, 2014.

[6] L. S. A. Armando Guzmán, and C. Labuschagne, "Adaptive Inverse-Time Elements Take Microprocessor-Based Technology Beyond Emulating Electromechanical Relays," 68th Annual Georgia Tech Protective Relaying Conference, Georgia, April 2014.

[7] G. Benmouyal, M. Meisinger, J. Burnworth, W. A. Elmore, K. Freirich, P. A. Kotos, et al., "IEEE standard inverse-time characteristic equations for overcurrent relays," IEEE Transactions on Power Delivery, vol. 14, pp. 868-872, 1999.

[8] T. Instruments, "Stellaris LM4F232 Evaluation Board User's Manual," in http://www.ti.com/lit/ug/spmu272/spmu272.pdf, 2012.