NEWTON'S NUMERICAL METHOD FOR OPTIMIZATION: A SIMPLER APPROACH

Muhibul Haque Bhuyan
Department of Electrical and Electronic Engineering
Green University of Bangladesh
Mirpur 2, Dhaka 1216, Bangladesh

E-mail: muhibulhb@gmail.com

Abstract: *In this paper, a simpler approach of Newton's numerical method for optimization has been proposed. The first and second derivatives of the original function in the Newton's method have been replaced by the first and second order finite-divided difference formulas. A problem has been chosen and MATLAB program has been developed for finding the optimum value using both Newton's method and the proposed modified Newton's method. It has been observed that the proposed method produces the same result produced by the Newton's method and takes same number of iterations and same amount of execution time. The proposed method eliminates the need for finding the first and second derivatives of the original function and hence this approach is simpler.*

Keywords: *Optimization, Newton's method, finite-divided-difference formula.*

1. Introduction

In mathematics, optimization, or mathematical programming, refers to choosing the best element from some set of available alternatives. In the simplest case, this means solving problems in which one seeks to minimize or maximize a real function by systematically choosing the values of real or integer variables from within an allowed set. More generally, it means finding "best available" values of some objective function given a defined domain, including a variety of different types of objective functions and different types of domains. Problems in rigid body dynamics, design problems in aerospace engineering, utility maximization in microeconomics and its dual problem the expenditure minimization etc. are solved using optimization techniques. Another field that uses optimization techniques extensively is operations research. The existence of derivatives is not always assumed and many methods were devised for specific situations. The basic classes of methods, based on smoothness of the objective function, are combinatorial methods, derivative-free methods, first-order methods and second-order methods. There are many methods falling into these categories. Of them, Newton's method is a second-order method. There exist robust, fast numerical techniques for optimizing twice differentiable convex functions [1-6].

In mathematics, Newton's method is a well-known algorithm for finding the roots of the equations in one or more dimensions. It can also be used to find local maxima and local minima of functions by noticing that if a real number x^* is a stationary point of a function $f(x)$, then x^* is a root of the derivative $f'(x)$, and therefore one can solve for x* by applying Newton's method to $f'(x)$. The Taylor's expansion of $f(x)$ is given by equation (1).

$$f(x+\Delta x) = f(x) + f'(x)\Delta x + \frac{1}{2} f''(x)\Delta x^2 \qquad (1)$$

attains its extremum when Δx solves the linear equation (2).

$$f'(x) + f''(x)\Delta x = 0 \qquad (2)$$

Thus, provided that $f(x)$ is a twice-differentiable function and the initial guess x_0 is chosen close enough to x^*, the sequence (x_n) defined by equation (3).

$$x_{n+1} = x_n - \frac{f'(x_n)}{f''(x_n)},\ n \geq 0 \qquad (3)$$

Date of submission : 11. 02. 2012 Date of acceptance : 19. 07. 2012

will converge towards x^* .

The geometric interpretation of Newton's method is that at each iteration one approximates *f(x)* by a quadratic function around, and then takes a step towards the maximum/minimum of that quadratic function. If *f(x)* happens to be a quadratic function, then the exact extremum is found in one step.

Newton's method converges much faster towards a local maximum or minimum than gradient descent. In fact, every local minimum has a neighborhood N such that, if we start with $x_0 \in N$, Newton's method with step size γ = 1 converges quadratically (if the Hessian is invertible in that neighborhood). But sometimes, Newton's method may diverge based on the nature of the function and the quality of the initial guess [7].

The problem of Newton's method is that one needs to find the first and second derivative of the given function *f(x)*. But sometimes finding first and second derivatives may not be convenient and time consuming. Therefore, in this paper a modified optimization technique based on Newton's method has been developed. In this method, analytical derivatives have been replaced by numerical finite-divided difference formulas of first and second order. Since the centre finite-divided difference formula produces better results than the forward and backward finite-divided difference formulas, therefore, in this modified optimization method the former finite-divided difference formulas have been used. Numerical solutions of optimum value of a selected function from electrical engineering field have been produced by using both of Newton's method and the proposed method. For this purpose MATLAB programs have been developed. Simulation results produced by these two methods show that the proposed method takes less number of iterations and hence less execution time.

2. Proposed Method

In some cases, we need to find the numerical differentiation. In that case, we use the following numerical methods [7]:

1. Forward finite-divided difference formula
2. Backward finite-divided difference formula
3. Centered finite-divided difference formula.

The first and second derivatives of the forward finite-divided difference formula are given in equations (4).

$$f'(x_n) = \frac{f(x_{n+1}) - f(x_n)}{h}, \quad n \ge 0 \qquad (4a)$$

$$f''(x_n) = \frac{f(x_{n+2}) - 2f(x_{n+1}) + f(x_n)}{h^2}, \quad n \ge 0 \qquad (4b)$$

The first and second derivatives of the backward finite-divided difference formula are given in equations (5).

$$f'(x_n) = \frac{f(x_n) - f(x_{n-1})}{h}, \quad n \ge 0 \qquad (5a)$$

$$f''(x_n) = \frac{f(x_n) - 2f(x_{n-1}) + 2f(x_{n-2})}{h^2}, \quad n \ge 0 \qquad (5b)$$

The first and second derivatives of the centered finite-divided difference formula are given in equations (6).

$$f'(x_n) = \frac{f(x_{n+1}) - f(x_{n-1})}{2h}, \quad n \ge 0 \qquad (6a)$$

$$f''(x_n) = \frac{f(x_{n+1}) - 2f(x_n) + f(x_{n-1})}{h^2}, \quad n \ge 0 \qquad (6b)$$

The equations (4), (5) and (6) are used in equation (3) to find the optimum value (either maximum or minimum) of the given function, *f(x)*.

3. Problem Selection

In this study, a simple electrical circuit has been selected (as shown in Fig. 1) to find its maximum power delivered to the load from a dc voltage source. The expression of electrical power delivered to the load resistance, R_a can be expressed as in equation (7).

$$P_a = I_a^2 R_a \qquad (7)$$

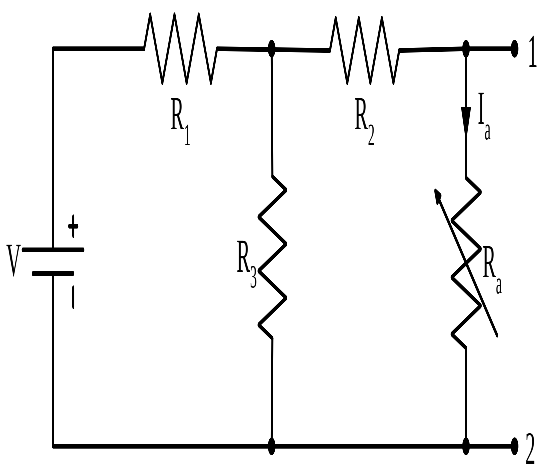

Fig. 1 An electrical circuit with an adjustable resistor R_a

The current through the load resistance can be calculated as follows

$$I_a = \frac{R_3}{R_3 + R_2 + R_a} \times \frac{V}{R_1 + R_3 \parallel (R_2 + R_a)}$$

$$= \frac{R_3}{R_3 + R_2 + R_a} \times \frac{V}{R_1 + \dfrac{R_3(R_2 + R_a)}{R_3 + R_2 + R_a}}$$

and is given by equation (8).

R_a varies. At some point of load resistance, the load power will be maximum. When the load resistance is equal to the network resistance looking in to the terminals 1-2 (as shown in Fig. 1) then the power transferred to the load will be maximum. The network resistance is actually the Thevenin's resistance, R_{th} at terminals 1-2, and it is obtained as follows:

$$R_{th} = R_1 \parallel R_3 + R_2$$

$$= \frac{R_1 R_3}{R_1 + R_3} + R_2$$

The value of resistances R_1, R_2 and R_3 are 10, 12 and 8 Ω respectively and the value of the supply voltage is taken as 80 V. Thus the calculated value of Thevenin's resistance, R_{th} is 16.44 Ω. Therefore, initial guess (R_{a0}) is selected as 4 Ω because the optimum value will be R_{th}. Step size (h) for the different methods is taken as 0.01.

To calculate the maximum (i.e. the optimum) load power using the Newton's method of optimization, first and second derivatives of the original expression of the load power given in equation (9) have been found out by differentiating equation (9) twice with respect to R_a. The first derivative of the load power, P_a' is given in equation (10) and the second derivative of the load power, P_a'' is given in equation (11).

$$P_a = \left[\frac{VR_3}{R_1(R_2 + R_3) + R_2 R_3 + R_a(R_1 + R_3)} \right]^2 R_a \quad (9)$$

$$P_a' = \left[\frac{VR_3}{R_1(R_2 + R_3) + R_2 R_3 + R_a(R_1 + R_3)} \right]^2 - \frac{2(VR_3)^2 (R_1 + R_3) R_a}{\left[R_1(R_2 + R_3) + R_2 R_3 + R_a(R_1 + R_3) \right]^3} \quad (10)$$

$$P_a'' = -\frac{4(VR_3)^2 (R_1 + R_3)}{\left[R_1(R_2 + R_3) + R_2 R_3 + R_a(R_1 + R_3) \right]^3} + \frac{6(VR_3)^2 (R_1 + R_3)^2 R_a}{\left[R_1(R_2 + R_3) + R_2 R_3 + R_a(R_1 + R_3) \right]^4} \quad (11)$$

$$I_a = \frac{VR_3}{R_1(R_2 + R_3) + R_2 R_3 + R_a(R_1 + R_3)} \quad (8)$$

Therefore, the load power expression can be written in terms of resistances and applied voltages as in equation (9) from equations (7)-(8). The load power, P_a varies as load resistance,

4. Program Development in MATLAB

Three programs are developed in MATLAB for solving the optimum value of the function in (9) using Newton's method of optimization. One is the main file and the other two are the function files. One function file is for the first derivative

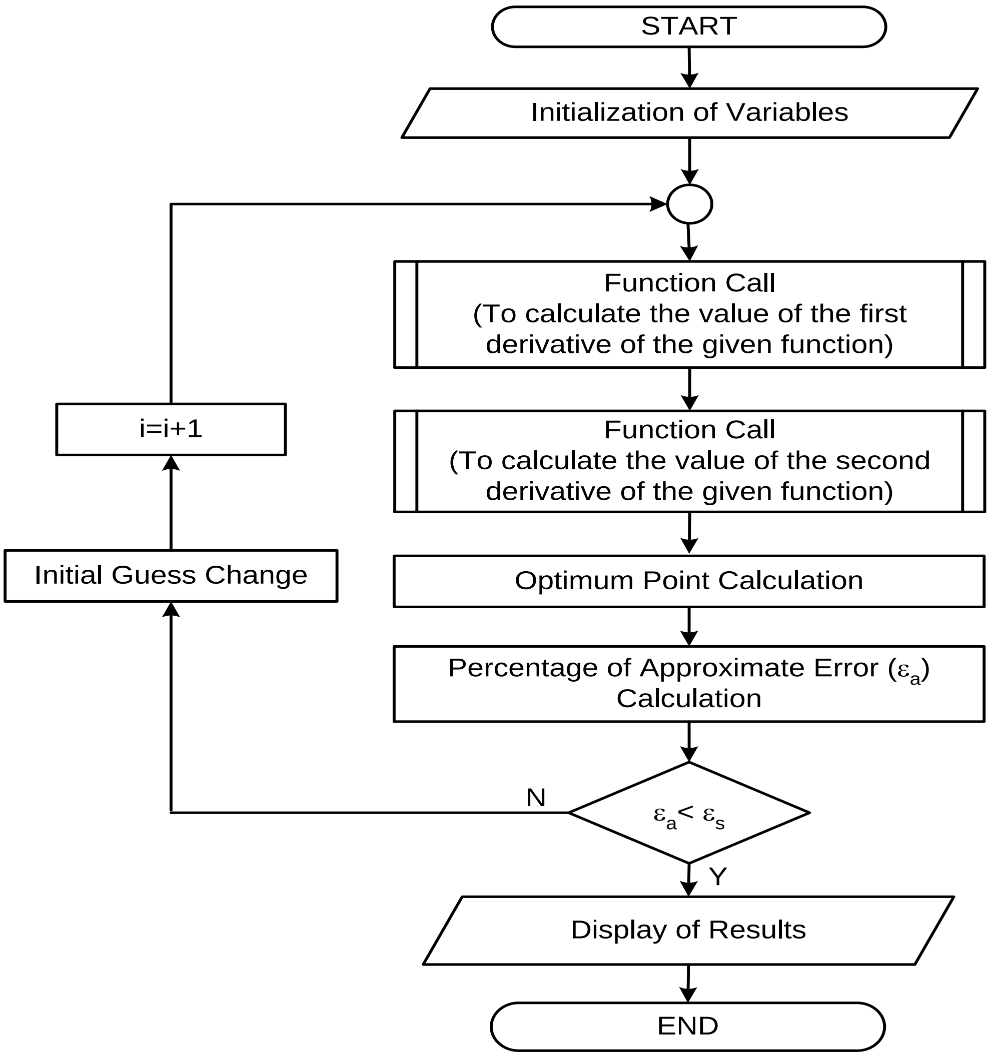

Fig. 2 Flow chart of the program developed in MATLAB for the optimum value calculation using the Newton's method of optimization

of the function and the other function file is for the second derivative of the function. But for solving the optimum value using the proposed method only two programs are developed. One is the main file and the other is the function file.

The flow charts of the main programs for the optimum value calculation using the Newton's method and the proposed methods of optimization are shown in Figs. 2-3. In the main file, at first initial guesses and other parameter values are declared. Then the main file enters into a loop where algorithm of a particular method of optimization is written as shown in the flow charts of Figs. 2-3. Fig. 2 calls two different function files but Fig. 3 calls one function file two times. Optimum value is calculated according to the algorithm of a particular optimization method and then the percentage of approximate error (ε_a) is calculated. The absolute value of ε_a is compared to the absolute value of the specified error limit (ε_s). If the absolute value of ε_a is greater than the absolute value of ε_s then the method does not converge yet and iteration is continued after changing the initial guess for the next iteration. If the absolute value of ε_a becomes less than the absolute value of ε_s then the method converges, iteration is stopped and finally, the results are

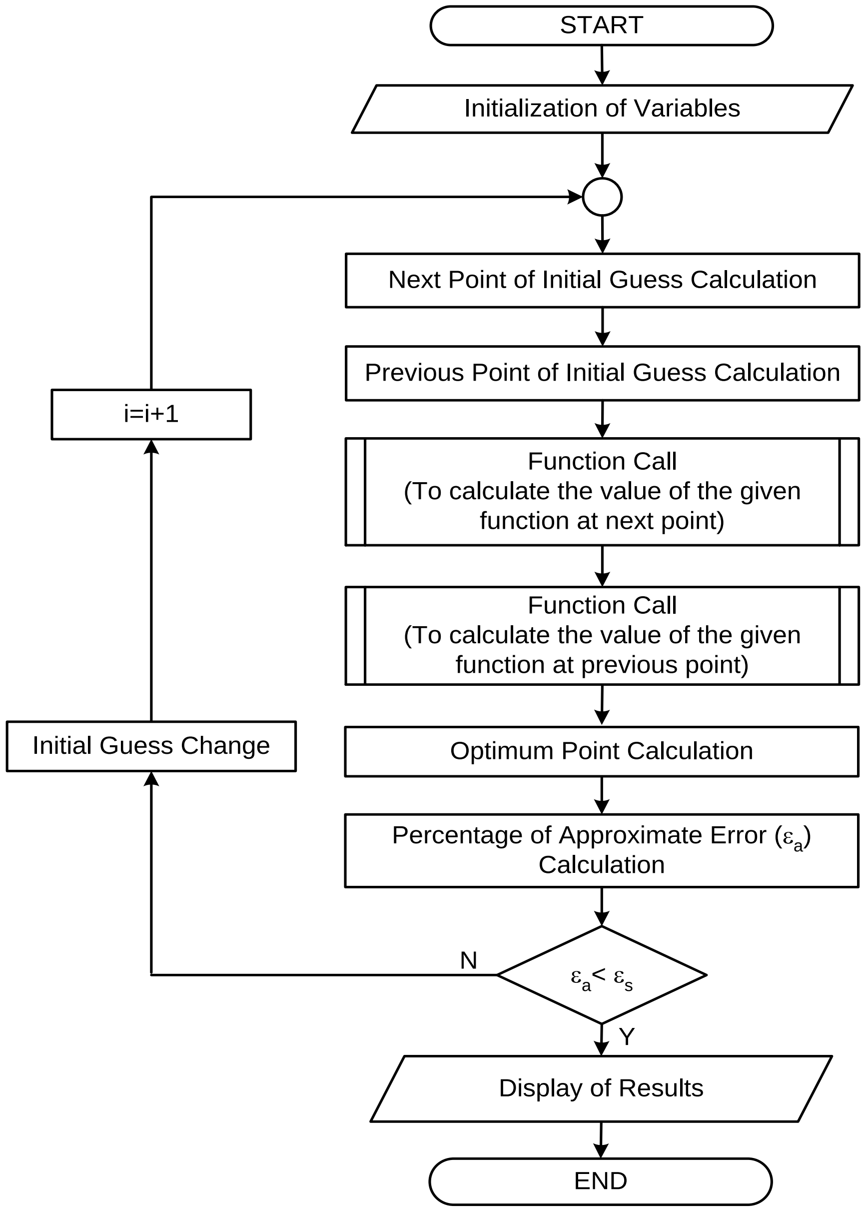

Fig. 3 Flow chart of the program developed in MATLAB for the optimum value calculation using the proposed method of optimization

displayed at the command prompt of MATLAB. Execution time of the each program is also displayed at the command prompt of MATLAB. Besides, the number of iterations required to calculate the optimum values and percentage of approximate errors are saved in a text file for the analysis of the results and comparison of the different methods.

The function file is written in a separate text file using the MATLAB command, 'function' followed by the variable where calculated value of the function is saved and returned to the main file. Then function name is inserted by which the main file calls the function file and the argument of the function is given in parenthesis through which the main file sends the value of the function's unknown variable. This first line is known as function declaration as in Fig. 4. After that, this file initializes all the parameters required for calculation of the function value for the obtained value of the argument variable from the main file. The main file calls this function file each time for calculation of the function values. In MATLAB, function file should be named by the function name. In the third step of the program, function value is calculated for the given function in (9). For the Newton's method, first and second derivative of the given function (given in (10) and (11)) should be used in the last line of the flow chart given in Fig. 4 instead of the original function in (9).

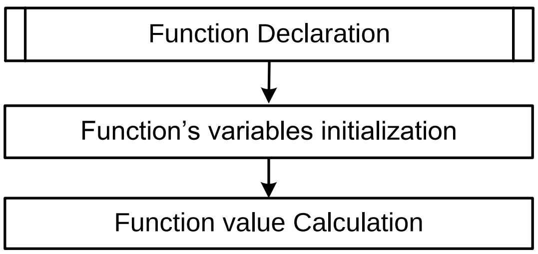

Fig. 4 Flow chart of the 'function' file developed in MATLAB for the optimum value calculation using the proposed method of optimization and also for the Newton's method of optimization

5. Simulation Results

MATAB simulations have been performed for the Newton's method and also the proposed method of optimization for the same function and same initial guess provided in the previous section. For this purpose, an IBM machine with 500 GB hard disk drive, 2 GB RAM and an Intel Core™ i3 CPU with 3.06 GHz clock frequency is used. Microsoft Windows XP is used as the operating system, and MATLAB program runs under this operating system. Built-in timers of MATLAB have been used to calculate the execution time of each program. The MATLAB simulation results are show in Table 1. From MATLAB simulation, it has been observed that the Newton's method takes 0.038242 sec, new method with forward finite difference formula takes 0.035214 sec, new method with backward finite difference formula takes 0.035214 sec and new method with centered finite difference formula takes 0.035214 sec for the execution of the program. That is, the proposed new methods take almost the same amount of execution time in the MATLAB environment. So, if execution time is considered, any one of the proposed methods can be chosen instead of the Newton's method.

From Table 1, it is observed that the Newton's method and the new methods converge after the seventh iteration for the same value of the percentage of specified error (ε_s). But the new method using the centered finite-divided difference formula produces the least amount of percentage of approximate errors while the backward finite-divided difference formula produces the highest amount of percentage of approximate errors. Besides, from the table it is observed that the new method using the centered finite-divided difference formula produces the final result, i.e. the optimum point that is very close to the result produced by the Newton's method at the same value of ε_a.

Table 1 Number of iterations, optimum point, maximum value and percentage of approximate errors for different methods

Iterations	Newton's Method			New approach using forward finite-divided difference formula		
	$R_{a,op}$	P_{max}	ε_a	$R_{a,op}$	P_{max}	ε_a
1	8.4034188	26.88517548	52.40033	8.40855835	26.89049326	52.42941972
2	12.48343656	29.46699692	32.68345	12.4908984	29.46940549	32.68251744
3	15.29110686	29.99036785	18.36146	15.29527639	29.99066445	18.33492852
4	16.3310619	30.02967065	6.367957	16.32874436	30.02965576	6.32913314
5	16.44327983	30.03002999	0.682455	16.43843485	30.03002903	0.66728062
6	16.44444432	30.03003003	0.007081	16.43944603	30.03002934	0.00615095
7	16.44444444	30.03003003	7.5×10^{-7}	16.4394452	30.03002934	5.05×10^{-6}

Iterations	New approach using backward finite-divided difference formula			New approach using centered finite-divided difference formula		
	$R_{a,op}$	P_{max}	ε_a	$R_{a,op}$	P_{max}	ε_a
1	8.398279	26.87985	52.37119	8.40342	26.88518	52.400334
2	12.47597	29.46458	32.68434	12.48344	29.467	32.683454
3	15.28691	29.99007	18.38791	15.29111	29.99037	18.361459
4	16.33335	30.02969	6.406779	16.33107	30.02967	6.3679535
5	16.44812	30.03003	0.697771	16.44328	30.03003	0.6824521
6	16.44944	30.03003	0.00805	16.44445	30.03003	0.0070811
7	16.44945	30.03003	8.31×10^{-6}	16.44445	30.03003	7.5×10^{-7}

$R_{a,op}$ = Optimum value of the load resistance (Ω)
P_{max} = Maximum power delivered to the load resistance (W)
ε_a = Percentage of approximate errors (%)

6. Conclusions

A simpler method of optimization has been proposed in terms of the Newton's method of optimization. The proposed method and the Newton's optimization method are solved for a particular problem by developing MATLAB programs. The results are then represented in tabular form. It is observed that the new method takes less execution time. Besides, new method produces more error than the Newton's method of optimization for the same number of iterations as well as for the same amount of specified error limit. It is suggested that the new method using the centered finite-divided difference formula should be used as an alternative of the Newton's method of optimization. As future extensions of this work more practical problems can be solved using the proposed methods and results can be analyzed to observe the suitability of the proposed method in different branches of engineering disciplines.

References

[1] Mordecai Avriel, "Nonlinear Programming: Analysis and Methods," Dover Publishing, ISBN 0-486-43227-0, 2003.

[2] Stephen Boyd and Lieven Vandenberghe, "Convex Optimization," Cambridge University Press, ISBN 0-521-83378-7, 2004.

[3] K. H. Elster, "Modern Mathematical Methods of Optimization," Vch Publication, ISBN 3055014529, 1993.

[4] Jorge Nocedal and Stephen J. Wright, "Numerical Optimization", Springer, ISBN 0-387-30303-0, 2006.

[5] Panos Y. Papalambros and Douglass J. Wilde, "Principles of Optimal Design: Modeling and Computation," Cambridge University Press, ISBN 0-521-62727-3, 2006.

[6] X. S. Yang, "Introduction to Mathematical Optimization: From Linear Programming to Metaheuristics," Cambridge International Science Publishing, ISBN 1-904602827, 2008.

[7] Steven C. Chapra and Raymond P. Canale, "Numerical Methods for Engineers with Software and Programming Applications," 4th Edition, ch. 1, Tata McGraw Hill Publishing Co. Ltd., India, 2002.

Dr. Muhibul Haque Bhuyan born in Dhaka, Bangladesh in 1974. He received the B.Sc. Engg., M.Sc. Engg. and Ph.D.

degrees all in Electrical and Electronic Engineering (EEE) from Bangladesh University of Engineering Technology (BUET), Dhaka, Bangladesh in 1998, 2002 and 2011 respectively.

At present, Dr. Bhuyan is working as an Associate Professor and Chairman in the Department of Electrical and Electronic Engineering (EEE) of Green University of Bangladesh (GUB) since July 2012. He started his career as a Faculty Member in the Department of ECE/EEE under the Faculty of Engineering of American International University Bangladesh (AIUB) and worked there from June 1999 to June 2003. He was with the Ultra-Scaled Devices Engineering Laboratory under the Center of Excellence (COE) program of Hiroshima University, Higashi-Hiroshima, Japan as a Researcher from July 2003 to March 2004. From July 2005 to August 2008, Dr. Bhuyan led the Department of Electronics and Telecommunication Engineering (ETE) of Daffodil International University (DIU) as Head of the Department and played a pioneering role in establishing the Department of ETE and in opening a new Department of Electrical and Electronic Engineering (EEE) in DIU. He worked as an Assistant Professor in DIU from July 2005 to May 2010 and in Presidency University (PU) from June 2010 to December 2011 and as an Associate Professor of the Department of EEE in DIU from January 2012 to June 2012. He also worked as a Part-time Lecturer in the Department of EEE of Ahsanullah University of Science and Technology, Bangladesh and as an Adjunct Assistant Professor in the Department of CSE/ETE/EEE of East West University (EWU), Peoples University of Bangladesh (PUB) and DIU. His research interest includes modeling of the pocket implanted bulk n-MOSFET, double-gate and SOI n-MOSFET, power electronics and control systems design using PLC, microcontroller, DSP, FPGA etc. He is involved in numerical computations of electrical and electronic engineering problems using MATLAB. He also works on the development and improvement of electrical and electronic engineering education in private university sector in Bangladesh. He has around 50 national and international journal and conference papers.

Dr. Bhuyan was the Secretary of IEEE Bangladesh Section. He is the Executive Member of Bangladesh Electronics Society (BES), Life Member of Institute of Engineers Bangladesh (IEB) and Member of IEEE and EDS.