

# ESTIMATION OF OPTIMAL TIME STEP AND COMPARE WITH THE ODE SOLVER OF MATLAB PACKAGE OF RLC CIRCUIT USING NUMERICAL METHODS

Sheikh Md. Rabiul Islam

Department of Electronics & Communication Engineering  
Khulna University of Engineering & Technology, Khulna, Bangladesh.

Email: robi\_kuet@yahoo.com; robi\_ece@ece.kuet.ac.bd

**Abstract:** In this paper analysis of a RLC circuit model that has been described optimal time step and minimize of error using numerical method. The goal is to reach the optimal time response due to the input for which optimal output response reaches a minimum error and also compared with ODE solver of MATLAB packages for the different cell (mesh) size of the RLC model. Table is constructed of the model to evaluate optimal time step and also CPU time into the simulation using MATLAB 7.6.0(R2008a). The values of register, capacitor and inductor as well as electromagnetic force are obtained through the mathematical relations of the model. The general analysis of the RLC circuit due to the optimal time step and minimum error is developed after several analysis and operations. The theoretical results show effectiveness of optimized of the model.

**Keywords:** Optimal time step, MATLAB, Trapezoidal, Implicit Euler, Runge-Kutta method and RLC circuit.

## 1. Introduction

In practice, engineering problems are difficult to solve. Most often, numerical methods are used as analytical solutions to such problems may be non-existent. Numerical methods in themselves are usually iterative in nature requiring several intermediate steps in order to arrive at a solution. An RLC circuit (also known as a resonant circuit or a tuned circuit) is an electrical circuit consisting of a resistor (R), an inductor (L), and a capacitor (C), connected in series or in parallel. An RLC circuit is called a *second-order* circuit as any voltage or current in the circuit can be described by a second-order differential equation for circuit analysis. One very useful characterization of a linear RLC circuit is given by its Transfer Function, which is (more or less) the frequency domain equivalent of the time domain input-output relation. These methods do not use any knowledge of the interior structure of

the plant, and as we have seen allows only limited control of the closed-loop behavior when feedback control is used.

In principle, since numerical methods and considered to solve the problem are firm, it would be possible to obtain an approximation error arbitrarily small simply by reducing the time step  $h$ . In a reality, however, as it is to stage and used in a defined number of bits, or that we can observe or to decline to that the relative error approximately haul, not to reach a minimum, and then begins to grow because of the distortions introduced by a precisely defined representation of the computer. Calculate the optimal time step significant to find an  $h$  for which the relative error is minimum. The operation can be completed only with a series of experiments using different  $h$ . One consideration that is or can do and that in a near optimal time step of the order of the method. Therefore, the MATLAB program used to calculate the first order by comparing the errors obtained using mesh with a ratio of  $h1$  and  $h2$  constant and equal to two (i.e. doubling at each iteration and the number of steps) and then, when the order obtained experimentally haul at least 10% compared to the expected value is the internal a divider between the last two values used in five and is calculated to the error for these values of  $h$ . The value of which will have to associate the error becomes a minimum so the optimal time step. The decision to divide the interval into five and arbitrary and the result of a trade-off between trying to value can close the range to an absolute minimum and the computational burden.

In this paper, we have consider series RLC-circuit we have a resistor R (ohms), an inductor L (henrys), a capacitor C (farads) and an source of electromotive source  $E(t)$  and the following

voltage drops:

$$E_L = LI' ; \text{ across the inductor}$$

$$E_R = RI ; \text{ across the resistor}$$

$$E_C = \frac{1}{C} \int I(t) dt ; \text{ across the capacitor}$$

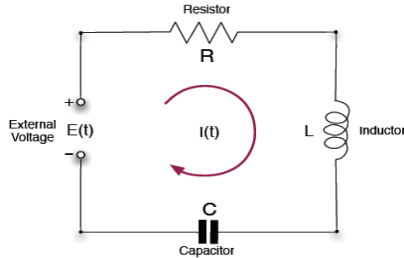


Fig. 1: RLC circuit

Using Kirchhoff's law one obtains the second order linear ordinary differential equation

$$I'' + \frac{R}{L} I' + \frac{1}{LC} I(t) = \frac{1}{L} E(t) \quad (1)$$

Consider the following situation:

$$E(t) = E_0 \sin(\omega t) \text{ Electromotive force}$$

$I(0) = 0$  initial condition at time 0 and the following parameters

$$R = 100 \text{ ohms}$$

$$L = 0.1 \text{ henry}$$

$$C = 10^{-3} \text{ farad}$$

$$\omega = 377$$

$$E_0 = 155$$

The exact solution is given by

$$I(t) = C_1 e^{-10.10205140t} + C_2 e^{-989.8979486t} + C_3 \cos(377t) + C_4 \sin(377t)$$

(2)

The authors described it for the explicit methods: second-order Runge-Kutta, and fourth-order Runge-Kutta, in the time interval  $[0, 1]$  and compare results with the exact solution. The problem is solved using some "package" inside the MATLAB and compare with the exact solution and explicating CPU times. For each method the time step that gives a maximum absolute error  $e = 10^{-6}$  are found, the CPU times and which method is the most efficient is discussed.

In this paper, different numerical methods have been discussed in the following sections that represents the survey of numerically with MATLAB package for the RLC circuit in telecommunication system. Section IV concludes the results.

## 2. Numerical Method Analysis for RL Circuit Model

### 2.1 The Explicit Euler Method

Explicit Euler's method [1] is the simplest case of a Taylor method, where only the first term of the increment function is used, with second and higher order terms neglected.

The method is as follows:

$$I^{n+1} = I^n + h s(t^n, I^n) \quad (4)$$

Where,  $s(t^n, I^n)$  is the source term and

$$s(t^n, I^n) = \frac{R}{L} I^n + \frac{E_0}{L}$$

Substituting the values imposed by the problem, we obtain the following source term:

$$s(t^n, I^n) = I^n + 1 \quad (5)$$

Substitute the equation (4) in:

$$I^{n+1} = I^n + h(I^n + 1) \quad (6)$$

The initial condition was not specified by the problem. The value  $I_0 = 0$ , has been chosen and because this choice is equivalent to consider a circuit with a switch located between the source and the resistance ended instantly  $t_0 = 0$  and it seemed the most reasonable.

Note that the exact solution and the specific catalyst problem. Substituting the numerical values of parameters of the circuit, it is given by:

$$I(t) = e^{-t} + 1 \quad (7)$$

The explicit Euler's method is very simple to use but accuracy can get only first-order solution.

### 2.2 The second order Runge-Kutta method

We know that Runge-Kutta method is a second order accurate method. The method is as follows[1]:

$$I^{n+1} = I^n + h\varphi(t^n, I^n, h) \quad (8)$$

$$\varphi(t^n, I^n, h) = \frac{1}{2(k_1 + k_2)}$$

$$k_1 = s(t^n, I^n)$$

$$k_2 = s(t^n + h, I^n + hk_1)$$

For practical implementation, at each time step, it is required to calculate the coefficient  $k$  then run the value of  $I^{n+1}$ .

### 2.3 The Fourth Order Runge-Kutta method

This is a popular higher order numerical method [1]. in particular, it is a fourth order accurate method whose scheme is:

$$I^{n+1} = I^n + h\varphi(t^n, I^n, h)$$

$$\begin{aligned} \varphi(t^n, I^n, h) &= \frac{1}{2(k_1 + 2k_2 + 2k_3 + k_4)} \tag{9} \\ k_1 &= s(t^n, I^n) \\ k_2 &= s(t^n + h, I^n + hk_1) \\ k_3 &= s\left(t^n + \frac{1}{2}h, I^n + \frac{1}{2}hk_2\right) \\ k_4 &= s(t^n + h, I^n + hk_3) \end{aligned}$$

**2.4 Euler Implicit**

The scheme of this method is  $I^{n+1} = I^n + hs(t^{n+1}, I^{n+1})$  and said to be implicit [1] because the unknown  $I^{n+1}$  appear also as argument of the function  $s(t^n, I^n)$  which in general is a non-linear function. This method is similar as describe in A, and in particular it's always a method of first order. An equation of this type must be solved iteratively, meaning a greater computational effort comparing with previous methods. An implicit method is mandatory in order to solve stiff problems. A general method for solving the implicit Euler has also been developed, using Newton iteration for ordinary differential equation (1).

**2.5 The Implicit Trapezoidal Method**

The implicit trapezoidal method [1] is a second order accurate method whose scheme is

$$I^{n+1} = I^n + \frac{h}{2[s(t^n, I^n) + s(t^{n+1}, I^{n+1})]} \tag{10}$$

Also, in this case, it has been implemented in two ways, Newton iteration which is implemented for ordinary differential equation (1).

**3. The Model of RLC Circuit Analysis**

For the simulation numerically the equation (1) can be written in the form for needed to analysis:

$$\begin{bmatrix} I(t) \\ I'(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{1}{LC} & \frac{R}{L} \end{bmatrix} \begin{bmatrix} I(t) \\ I'(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{LC} E'(t) \end{bmatrix} \tag{4}$$

It has to be noticed the exact solution is not fully congruent with the initial condition for the current  $I(t)=0$  by inserting  $t=0$  the result  $I(0) = -0.006$  and also the derivative  $I'(t) = 5.88$ .

This value has been kept and inserted the MATLAB routines. For the scope of this paper it has not paid attention at the precision and correctness of the exact solution, but only at the relation between exact solution and numerical computed solutions. Note also that in the paper presentation there is an error for the modeling equation the right one is (4), with a derivative on the electromotive force. A graphical representation of the exact solution is shown in Fig. 1. The comparison of the exact solution and numerical solution are described in section A & B.

**3.1 Runge-Kutta 2<sup>nd</sup> order and Runge-Kutta 4<sup>th</sup> order**

In this paragraph the second and fourth order Runge-Kutta methods [1];[2] are simulated. A graph of entire solution is not useful, because they completely overlap, but a better discussion can be made by observing the maximum absolute error achieved in each simulation. The error is found by comparison with exact solution.

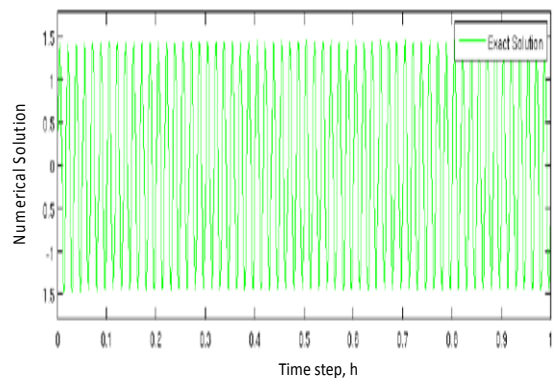


Fig. 1: Illustration of the exact solution for the RLC circuit.

Table 1: Summarized Results about the Optimal Time Steps for Range Kutta 2<sup>nd</sup> and Runge Kutta 4<sup>th</sup> Order Methods

| 2 <sup>nd</sup> order RungeKutta  |        |              | 4 <sup>th</sup> order RungeKutta  |        |              |
|-----------------------------------|--------|--------------|-----------------------------------|--------|--------------|
| Error                             | Cells  | Time Steps   | Error                             | Cells  | Time Steps   |
| 8.528778E-04                      | 10000  | 1.000000E-04 | 8.088048E-04                      | 10000  | 1.000000E-04 |
| 8.089193E-04                      | 50000  | 2.000000E-05 | 8.087351E-04                      | 50000  | 2.000000E-05 |
| 2.853225E-03                      | 100000 | 1.000000E-05 | 8.087350E-04                      | 100000 | 1.000000E-05 |
| 8.087356E                         | 500000 | 2.000000E-06 | 8.087349E-04                      | 500000 | 2.000000E-06 |
| 8.087351E-04                      | 1e+06  | 1.000000E-06 | 8.087350E-04                      | 1e+06  | 1.000000E-06 |
| <b>Running time: 87.0413 sec.</b> |        |              | <b>Running time: 169.632 sec.</b> |        |              |
| Minimum Error                     | Cells  | Time step    | Minimum Error                     | Cells  | Time step    |
| 8.087351E-04                      | 1e+06  | 1.000000E-06 | 8.087349E-04                      | 500000 | 2.000000E-06 |

From Table 1 it is possible to note that the 4<sup>th</sup> order Runge-Kutta method is better accurate in the first simulation, but when the mesh (cells) size increased the 2<sup>nd</sup> order Runge-Kutta reached the same accuracy as the fourth order one. For each simulation performed the error is expressed in maximum absolute value.

**3.2 Computing of the optimal time step for Runge-Kutta 2<sup>nd</sup> and Range-Kutta 4<sup>th</sup> order methods**

The method of 2<sup>nd</sup> order Runge-Kutta and 4<sup>th</sup> order Runge-Kutta method are described in [1];[2]. In Table I has exposed the output of an algorithm for finding the optimal time step. Look carefully at the errors it has to be notices that they do not decrease, even with a sensible

increasing of the mesh (cells) size. Fig. 2 and 3 it can be concluded that the error converges to a specific value, and do not decrease. For this reasons it decided not to perform simulation at a more refined mesh.

**3.3 Implicit Euler and Trapezoidal Methods**

The method of Implicit Euler and trapezoidal method are described in [1]; [2].In Table. III are summarized the simulations results about the implicit Euler and trapezoidal methods. A comparison with exact solution can be shown in Fig. 4 only for graphical purposes, and it shows that the numerical method follow the exact solution without over/under-shoot but it only disjoint from them.

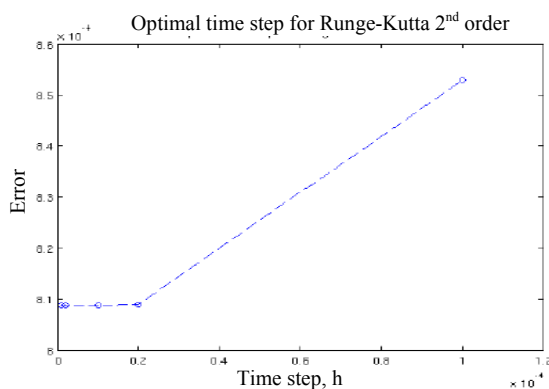


Fig. 2: Error behavior as the time step varies for Runge-Kutta 2<sup>nd</sup> order numerical method. The error has convergence to almost constant value.

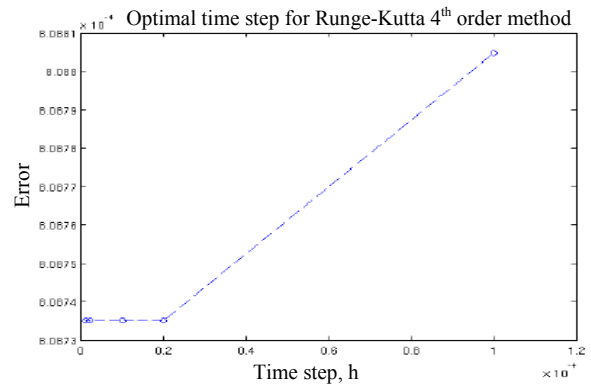


Fig. 3: Error behavior as the time step varies Runge-Kutta 4<sup>th</sup> order numerical method. The error has convergence to almost constant value.

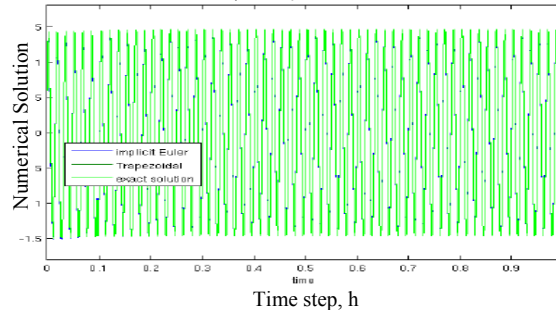


Fig. 4. The modeled difference equation simulated numerically by using implicit Euler method and Trapezoidal method. A mesh size of  $10^4$  cells has been used. Note that the methods graphically completely overlapped with the exact solution

Table 2: Summarized results about the implicit Euler and Trapezoidal methods

| Euler Implicit Newton |        |              | Trapezoidal Newton |        |              |
|-----------------------|--------|--------------|--------------------|--------|--------------|
| Error                 | Cells  | Time Steps   | Error              | Cells  | Time Steps   |
| 5.887045E-01          | 1000   | 1.000000E-03 | 2.519998E-02       | 1000   | 1.000000E-03 |
| 1.198974E-01          | 5000   | 2.000000E-04 | 9.096291E-04       | 5000   | 2.000000E-04 |
| 6.032776E-02          | 10000  | 1.000000E-04 | 6.331995E-04       | 10000  | 1.000000E-04 |
| 1.253281E-02          | 50000  | 2.000000E-05 | 8.011325E-04       | 50000  | 2.000000E-05 |
| 6.560170E-03          | 100000 | 1.000000E-05 | 8.068310E-04       | 100000 | 1.000000E-05 |
| Minimum Error         | Cells  | Time step    | Minimum Error      | Cells  | Time step    |
| 6.560170E-03          | 100000 | 1.000000E-05 | 6.331995E-04       | 10000  | 1.000000E-04 |

**4. Survey of Numerically with MATLAB Packages**

**4.1 Simulation Results using MATLAB Packages**

In Table IV are exposed the simulation results by using MATLAB packages. The following built in tools where used:

- Ode23 Runge-Kutta embedded of order 2 and 3 by Bogaci and Shampine.
- Ode45 Runge-Kutta embedded of order 4 and 5 by Dormand and Prince(first choice scheme)
- Ode15s Numerical Differentiation Formula, Similar to Backward Differentiation Formula by Gear. Useful for stiff problem.
- Ode23s Resenbrock of order 2.For stiff problem.
- Ode23t Trapezoidal rule for moderately stiff problem.
- Ode23tb TR-BDF” an implicit Rung-Kutta where the first stage is the trapezoidal

rule and the second stage is a BDF formula of order 2.

By comparing the CPU time of Table I&IV, It has to be noted that the comparison can be made due to the same mesh size adopted in both simulations, and by observing that each simulation belong to the same magnitude order, it can be concluded that the methods built in this paper are well optimized.

**4.2 Simulation Results at a Preset Error**

It has been choose an error  $e = 2 \times 10^{-4}$ , evaluated at the final time  $t = 1$ . The reason of this choice is due to the problem type and the experience about previous simulations, where the error is reached. The results are summarized here:

This briefing clearly shows that the most efficient method is the fourth order Runge-Kutta method as it is expected. Its higher accuracy coherent with its higher order.



Table 3: Summarized Results of Numerical Simulation from Built in MATLAB Packages

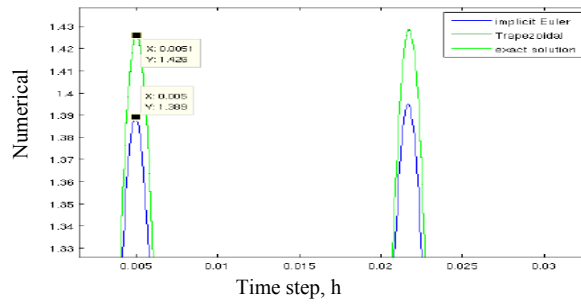


Fig. 5. The graph shows the higher precision of the Trapezoidal method, which is second order accurate, it graphically overlap with the exact solution.

(A)

| <i>ode23</i>                  |        |              | <i>ode45</i>                  |        |              |
|-------------------------------|--------|--------------|-------------------------------|--------|--------------|
| Error                         | Cells  | Time Steps   | Error                         | Cells  | Time Steps   |
| 2.337304E-03                  | 10000  | 1.000000E-04 | 9.110558E-04                  | 10000  | 1.000000E-04 |
| 2.337304E-03                  | 50000  | 2.000000E-05 | 9.113480E-04                  | 50000  | 2.000000E-05 |
| 2.337304E-03                  | 100000 | 1.000000E-05 | 9.114237E-04                  | 100000 | 1.000000E-05 |
| 2.337389E-03                  | 500000 | 2.000000E-06 | 9.008021E-04                  | 500000 | 2.000000E-06 |
| 2.337389E-03                  | 1e+06  | 1.000000E-06 | 9.237875E-04                  | 1e+06  | 1.000000E-06 |
| Running time: 78.9321 seconds |        |              | Running time: 80.8408 seconds |        |              |
| Minimum Error                 | Cells  | Time step    | Minimum Error                 | Cells  | Time step    |
| 2.337304E-03                  | 10000  | 1.000000E-04 | 9.008021E-04                  | 500000 | 2.000000E-06 |

(B)

| <i>ode15s</i>                |        |              | <i>ode23s</i>                 |        |              |
|------------------------------|--------|--------------|-------------------------------|--------|--------------|
| Error                        | Cells  | Time Steps   | Error                         | Cells  | Time Steps   |
| 6.754750E-03                 | 10000  | 1.000000E-04 | 2.119578E-03                  | 10000  | 1.000000E-04 |
| 6.754750E-03                 | 50000  | 2.000000E-05 | 2.120378E-03                  | 50000  | 2.000000E-05 |
| 6.754777E-03                 | 100000 | 1.000000E-05 | 2.145229E-03                  | 100000 | 1.000000E-05 |
| 6.754962E-03                 | 500000 | 2.000000E-06 | 2.121977E-03                  | 500000 | 2.000000E-06 |
| 8.684043E-03                 | 1e+06  | 1.000000E-06 | 2.122073E-03                  | 1e+06  | 1.000000E-06 |
| Running time: 175.57 seconds |        |              | Running time: 80.8408 seconds |        |              |
| Minimum Error                | Cells  | Time step    | Minimum Error                 | Cells  | Time step    |
| 6.754750E-03                 | 10000  | 1.000000E-04 | 9.008021E-04                  | 500000 | 2.000000E-06 |

(C)

| <i>ode23t</i>                 |        |              | <i>ode23tb</i>                |        |              |
|-------------------------------|--------|--------------|-------------------------------|--------|--------------|
| Error                         | Cells  | Time Steps   | Error                         | Cells  | Time Steps   |
| 2.596739E-03                  | 10000  | 1.000000E-04 | 2.055779E-03                  | 10000  | 1.000000E-04 |
| 2.599303E-03                  | 50000  | 2.000000E-05 | 2.058860E-03                  | 50000  | 2.000000E-05 |
| 2.599303E-03                  | 100000 | 1.000000E-05 | 2.058860E-03                  | 100000 | 1.000000E-05 |
| 2.704645E-03                  | 500000 | 2.000000E-06 | 2.147377E-03                  | 500000 | 2.000000E-06 |
| 2.842742E-03                  | 1e+06  | 1.000000E-06 | 2.120148E-03                  | 1e+06  | 1.000000E-06 |
| Running time: 77.7073 seconds |        |              | Running time: 80.8408 seconds |        |              |
| Minimum Error                 | Cells  | Time step    | Minimum Error                 | Cells  | Time step    |
| 2.596739E-03                  | 10000  | 1.000000E-04 | 2.055779E-03                  | 10000  | 1.000000E-04 |

**5. Conclusion**

The result of the MATLAB solution reached the more accurate method the Trapezoidal Newton method as compared to others numerical method in Table I & II. The resultant method shown smaller error & mesh size and its fast from a computational point of view as well as CPU time. An optimal time step of RLC model is constructed by the above method and compared with ODE solver of MATLAB package found minimum error of it. We could be decided RLC model supported both numerical method and ODE solver of MATLAB package with the effect of minimum error. The resultant of optimal time step is flexible in the sense that there was no restriction of on the desired values of register, capacitor and inductor and also electromagnetic sources for different value of mesh (cell ) size. Optimal time step of RLC model for different input value are also investigated in future.

**References**

[1] E. F. Toro, and Riemann, "Solvers and Numerical methods for fluid dynamics," Second edition, Springer 1999.

[2] Moosa M. Al-Mazeed, Elham A. AL. Juwaiser, Asma'a M Al-Mamoun "An optimal RLC realization of a general optimal transfer function due to step input," in *proc. IEEE ICECS-2003*, pp.1292-1295. <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=01301751>.

[3] Moosa M. Al-Mazeedi, "Eva'luation of optimal responses by routh tables using matlab software," in *Proc. 12" International Conference On Control Theory AndApplicariom*, Egypt, 2002, pp. 220-225.

[4] J. D. Lambert "Numerical methods for ordinary differential systems. The initial value problem," John Wiley and Sons, 1997.

[5] L. F. Shampine, I. Gladwell, and S. Thompson "Solving ODEs with MATLAB," Cambridge University Press, 2003.



**Sheikh Md. Rabiul Islam**

received the B.Sc.in Engg. (ECE) from Khulna University, Khulna, Bangladesh in December 2003, and M.Sc. in Telecommunication Engineering from the University of Trento, Italy, in October 2009 an also going an M.Sc in Engineering (EEE) under the Khulna University of Engineering & Technology, Khulna\_9203, Bangladesh. He joined as a Lecturer in the department of electronics and communication engineering of Khulna University of Engineering & Technology, Khulna, in 2004, where he is currently an Assistant Professor in the same department in the effect of 2008. He has published two Journal and four conferences in national and International (Specially USA) level. Since 2004 to 2006 he has acted as a visiting lecturer of ECE and CSE discipline in Khulna University, Bangladesh. His research interests include antennas and propagation, wireless communications & signal processing, and biomedical engineering. Mr. Islam is an Associate Member of Institute of Engineers Bangladesh (IEB) and Life member of Bangladesh Electronic Society (BES).